



Lexema-RPA

Роботизация
бизнес-процессов

Инструкция
пользователя

Lexema-RPA Studio



Инструкция пользователя

Lexema-RPA Studio

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	5
Назначение программы.....	5
Что такое RPA?	5
Что могут роботы?	5
Сколько нужно времени для создания роботов?	6
Области использования роботов.....	6
С чего начать	7
Часть I. ЗНАКОМСТВО СО СТУДИЕЙ.....	11
Глава 1. Интерфейс главного окна	11
Глава 2. Основы работы	22
Часть II. МОДУЛЬ «EXCEL».....	31
Глава 3. Основные моменты модуля	31
Глава 4. Набор полей в составе модуля	37
Глава 5. Описание действий, доступных в модуле	41
Глава 6. Модуль «Закрыть Excel-файл».....	75
Часть III. Модуль «Word».....	76
Глава 7. Основные моменты модуля	76
Глава 8. Действия модуля.....	79
Часть IV. МОДУЛЬ «КЛИКЕР» И МОДУЛИ ПОИСКА ИЗОБРАЖЕНИЙ.....	83
Глава 9. Модуль «Кликер»	83
Глава 10. Модуль «Поиск изображения»	92
Глава 11. Модуль «Поиск изображений»	94
Часть V. МОДУЛЬ «РАБОТА С ФАЙЛАМИ»	97
Глава 12. Основные моменты модуля	97
Глава 13. Описание действий, доступных в модуле	99
Часть VI. МОДУЛИ-ОПЕРАТОРЫ: «ЦИКЛ», «УСЛОВИЕ», «WHILE», «TRY-CATCH».....	106
Глава 14. Модуль «Условие»	106
Глава 15. Модуль «Цикл»	112
Глава 16. Модуль «While»	114
Глава 17. Модуль «Break».....	116
Глава 18. Модуль «Return»	117
Глава 19. Модуль «Try/Catch»	118
Часть VII. МОДУЛИ «ПОЧТОВЫЕ АГЕНТЫ».....	120

Глава 20.	Модуль «Отправить письмо»	120
Глава 21.	Модуль «Считать письма»	125
Глава 22.	Переместить письмо	132
Часть VIII.	МОДУЛЬ «РАБОТА С БРАУЗЕРОМ».....	134
Глава 23.	Основные моменты модуля	134
Глава 24.	Описание действий	136
Часть IX.	МОДУЛЬ «ИНТЕРФЕЙС».....	143
Глава 25.	Основные моменты.....	143
Глава 26.	Настройки окна.....	144
Глава 27.	Добавление элемента интерфейса	148
Часть X.	МОДУЛИ ДЛЯ РАБОТЫ С ТЕКСТОМ.....	155
Глава 28.	Модуль «Обработка текста»	155
Глава 29.	Модуль «Распознавание текста»	157
Глава 30.	Модуль «Распознавание текста ABBYY»	159
Часть XI.	ДРУГИЕ МОДУЛИ	161
Глава 31.	Модуль «Вычисления»	161
Глава 32.	Модуль «Запустить процесс»	163
Глава 33.	Модуль «Завершить процесс»	165
Глава 34.	Модуль «SQL».....	166
Глава 35.	Модуль «HTTP OData»	168
Глава 36.	Модуль «HTTP запрос»	170
Глава 37.	Модуль «FTP запрос».....	171
Глава 38.	Модуль «JSON».....	172
Глава 39.	Модуль «XML»	174
Глава 40.	Модуль «PDF».....	175
Глава 41.	Модуль «Пауза»	177
Глава 42.	Модуль «Прокси».....	177
Глава 43.	Модуль «Создать скриншот».....	177
Глава 44.	Модуль «Лог».....	178
Часть XII.	Оркестратор.....	179
Глава 45.	Общая информация.....	179
Глава 46.	Создание новой задачи	181
Глава 47.	Раздел «Роботы»	185

ВВЕДЕНИЕ

Руководство предназначено для людей, которые хотят научиться писать роботов в программе Lexema-RPA Studio. Вы познакомитесь со всеми инструментами студии и основными модулями, применяемыми для создания роботов.

Lexema-RPA Studio, далее именуемая просто студия, предоставляет пользователю возможность настраивать последовательность взаимодействий с разнообразными программами. Взаимодействие настраивается с помощью программных модулей, которые поставляются в составе студии. Каждый модуль представляет собой отдельный функционал, необходимый для выполнения конкретного действия, и механизмы для его настройки.

Назначение программы

Программа Lexema-RPA Studio предназначена для разработки программных роботов.

Что такое RPA?

Robotic process automation (или RPA) — это форма технологии автоматизации бизнес-процессов, основанная на метафорическом программном обеспечении роботов (ботов) или работников искусственного интеллекта. Роботы RPA используют пользовательский интерфейс для сбора данных и управления приложениями.

Более простыми словами, *RPA* — это программные роботы, которые эмулируют¹ действия человека, производимые им на компьютере.

Что могут роботы?

Роботы могут выполнять рутинные операции на компьютере вместо человека:

- вводить данные с клавиатуры, манипулировать мышью, считывать информацию с экрана, получать и обрабатывать почту, взаимодействовать с офисными программами;
- интегрировать данные из одной системы в другую без использования дополнительных средств.

¹ Эмуляция (англ. emulation) в вычислительной технике — комплекс программных, аппаратных средств или их сочетание, предназначенное для копирования (или эмулирования) функций одной вычислительной системы (гостя) на другой, отличной от первой, вычислительной системе (хосте) таким образом, чтобы эмулированное поведение как можно ближе соответствовало поведению оригинальной системы (гостя).

Сколько нужно времени для создания роботов?

Роботы создаются в простом графическом интерфейсе, и время, затраченное на робота, напрямую зависит от его сложности – простых роботов можно написать за пару часов, на более трудоемких может уйти несколько дней.

Области использования роботов

Роботы-дублёры (интеграторы)

- Перенос информации из одной системы в другую;
- Перезаполнение данных.

HR-роботы²

- Поиск вакансий по заданным критериям.

Роботы-операторы

- Регистрация и ввод данных по скан-копиям документов.

Роботы-консолидаторы

- Консолидация данных в отчет;
- Массовая отправка или печать отчетов.

Роботы-закупщики

- Мониторинг торговых площадок;
- Выгрузка информации о тендерах³;
- Отбор поставщиков на основании заданных условий.

Роботы-предавцы

- Проверка благонадежности контрагентов⁴;
- Формирование коммерческих предложений.

² HR (англ.яз., Human resources) — область управления персоналом.

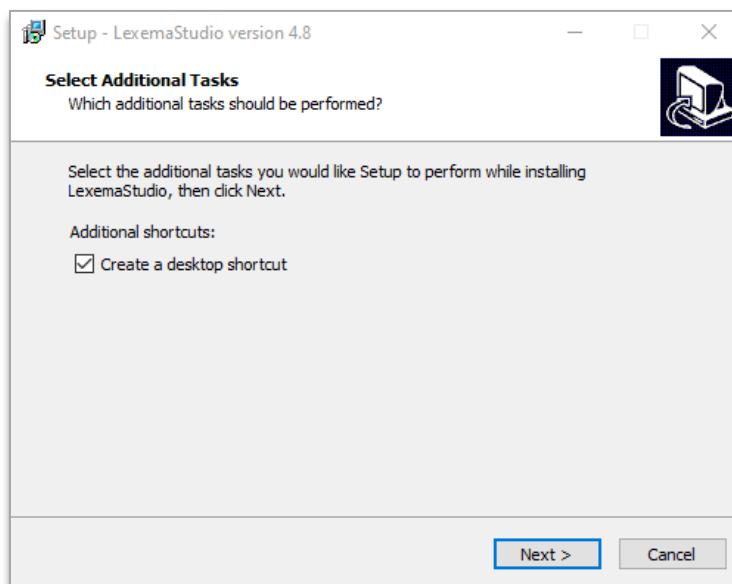
³ Тендер (англ. tender) — заявка на подряд, поданная на конкурсной основе.

⁴ Контрагент — лицо или учреждение, берущее на себя известные обязательства по договору.

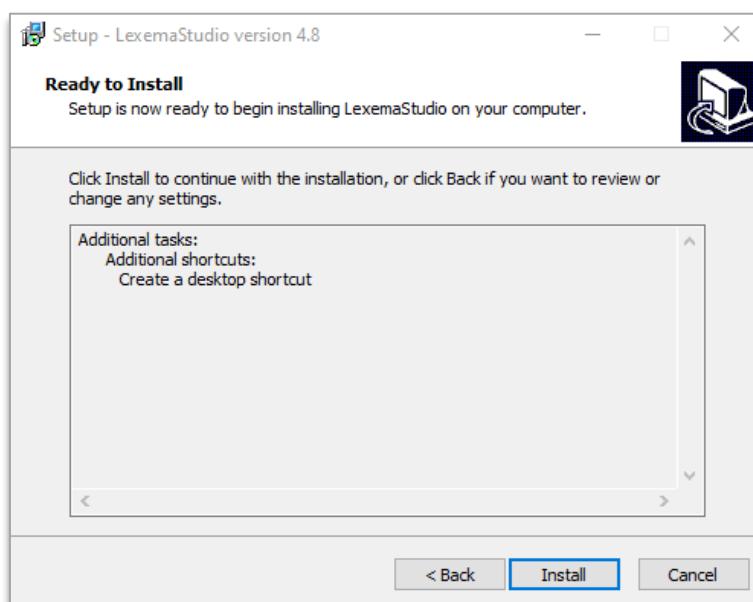
С чего начать

Для начала работы скачайте студию Lexema-RPA Studio *на сайте* <http://lexema.ru/lexemarpastudio/registratsiya/>. Перед началом скачивания вам будет предложена регистрация. Регистрация на сайте обязательна, так как доступ в студию осуществляется после авторизации пользователя. После окончания регистрации начнется скачивание установщика.

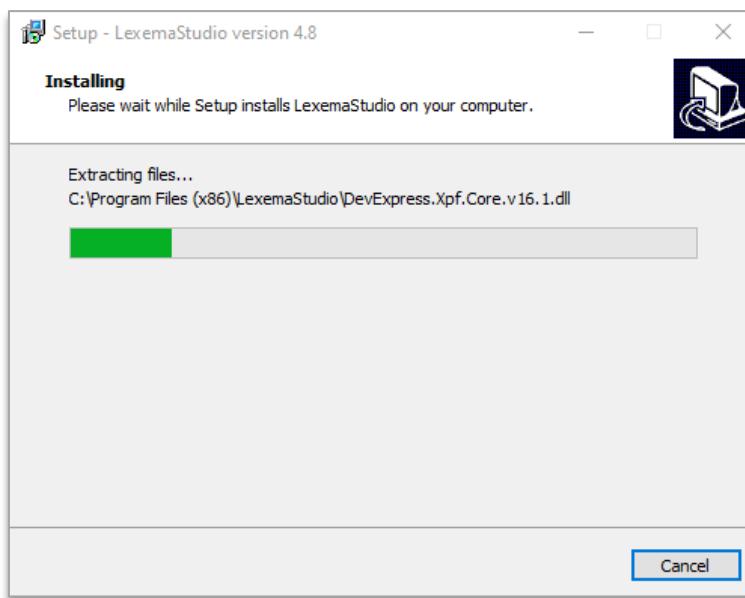
После этого запустите установщик и в открывшемся диалоговом окне нажмите на кнопку «Next».



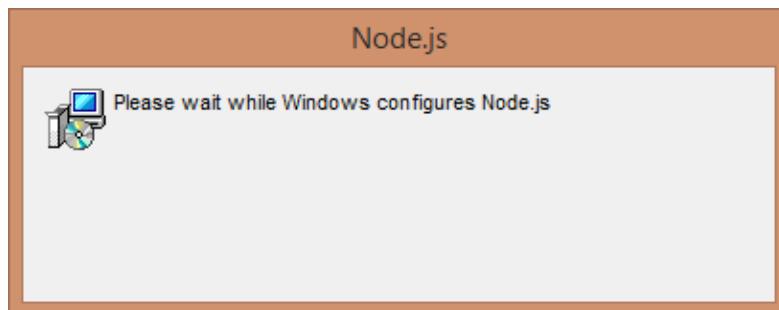
Затем нажмите на кнопку «Install».



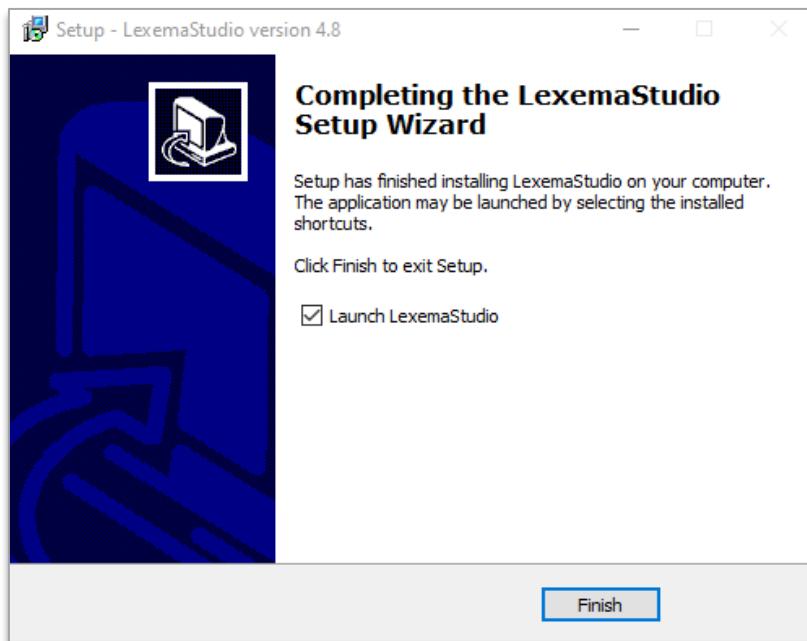
После этого начнется установка студии.



Затем начнется установка и конфигурация сервера Node.js, необходимого для работы студии.



Наконец, нажмите на кнопку «Finish».

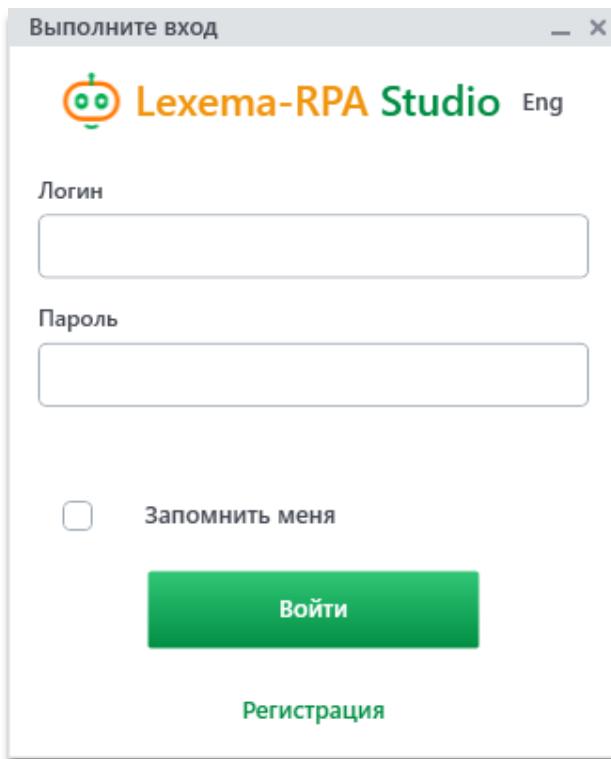


По завершению установки появится ярлык программы на рабочем столе.



СОВЕТ. Если у вас не получилось запустить или установить студию, установите (обновите) **.NET Framework 4.7.2** (Runtime версия). Сделать это можно по [ссылке](#) (Центр загрузки Windows).

После запуска программы появится окно авторизации, в которое следует ввести логин и пароль, указанные при регистрации. *Окно авторизации появляется в последних версиях программы.*



Флажок «Запомнить меня» запомнит ваши логин и пароль для последующих входов с текущего устройства.

Кнопка «Войти». По нажатию на кнопку произойдет вход в программу и откроется окно студии.

Если текущая версия программы не последняя, то перед авторизацией вам будет предложено скачать и установить обновление.

При нажатии на *Eng* справа от названия программы, программа перезапустится и весь язык интерфейса будет изменен на английский. Для перехода обратно на русский язык нужно будет проделать те же самые действия.

ЧАСТЬ I. ЗНАКОМСТВО СО СТУДИЕЙ

Глава 1. Интерфейс главного окна

Пользовательский интерфейс студии Lexema-RPA Studio характеризуется областью для работы с проектами (рабочей областью), расположенной в центре главного окна программы. Прямо над рабочей областью располагается полоса с кнопками управления проектом. Слева от рабочей области находится меню, в котором собраны модули для создания роботов, справа – обозреватель решений, прямо под рабочей областью – отладочный вывод.

Главное окно

Главное окно студии имеет несколько разделов и основные кнопки управления проектом (Рис. 1.1.1).

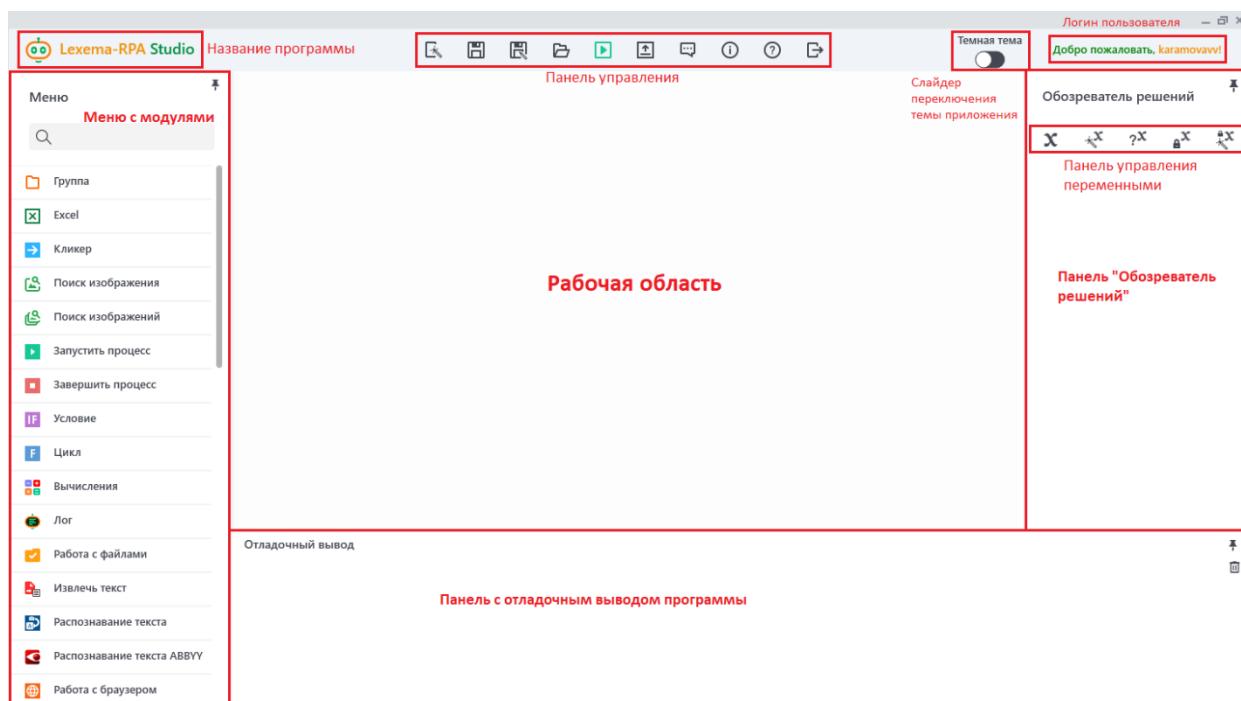


Рис. 1.1.1 Главное окно студии

Все панели (меню, обозреватель решений и отладочный вывод) могут быть закреплены или открываться при наведении на них (Рис. 1.1.2).

Чтобы открепить или закрепить панель необходимо нажать на иконку с изображением канцелярской кнопки  в правом верхнем углу панели.

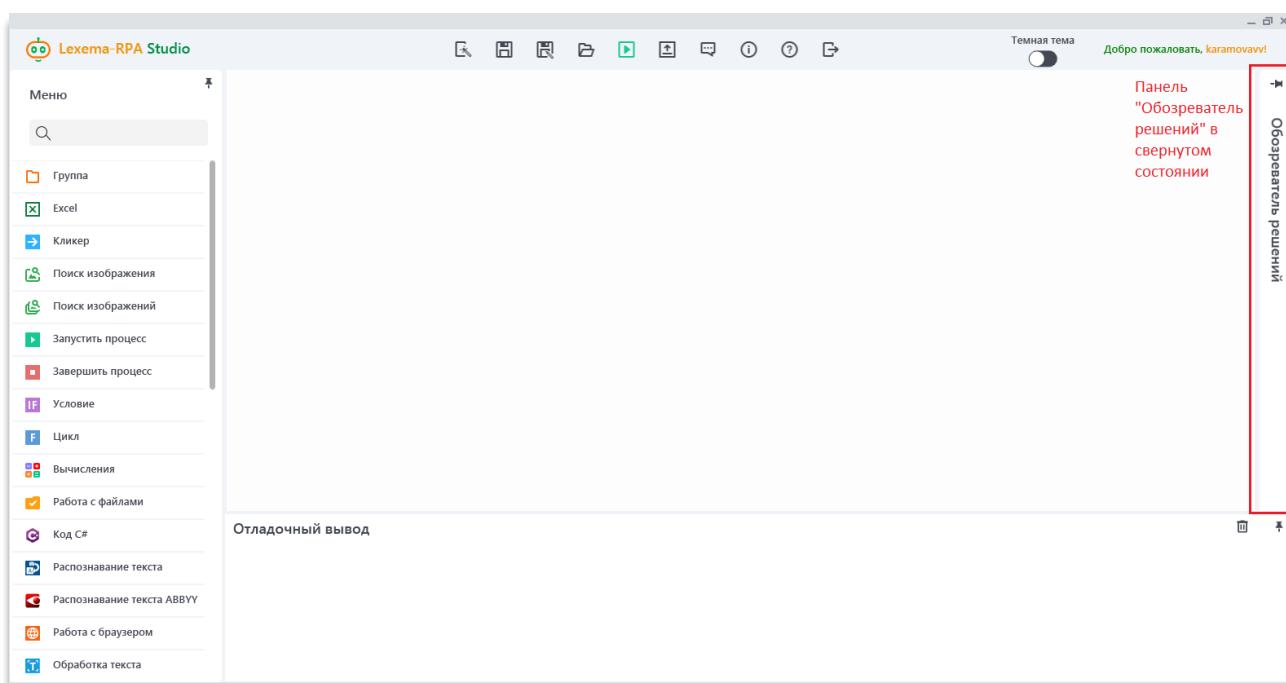


Рис. 1.1.2. Скрытая панель «Обозреватель решений»

Студия имеет две цветовые схемы – светлая тема, та, которая идет по умолчанию, и темная. Для перехода на темную тему необходимо нажать на слайдер «Темная тема», находящийся между кнопками управления и информацией о текущем пользователе.

Функционал разделов главного окна студии будет описан ниже.

Кнопки управления

Кнопки управления проектом находятся на верхней панели главного окна студии и предназначены для осуществления основных манипуляций с проектом: создания , сохранения , открытия , выполнения описанных в проекте действий  и выхода из программы .

Также поддерживаются различные комбинации горячих клавиш. Например, для создания проекта можно использовать комбинацию **Ctrl+N**.

Более подробно поддерживаемые комбинации горячих клавиш описаны в разделе «Горячие клавиши».

Создать

Нажатие на кнопку «Создать» вызывает окно создания нового проекта (Рис. 1.1.3).

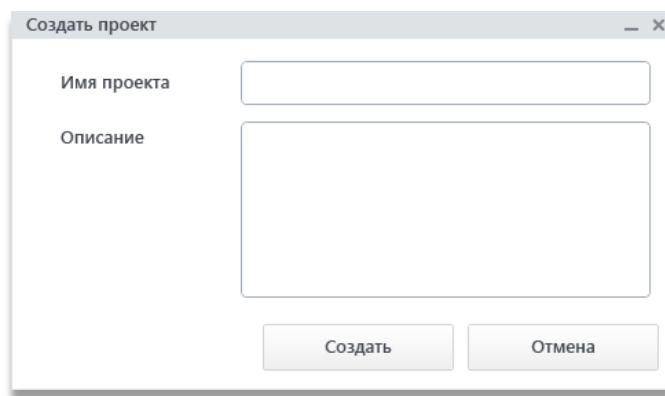


Рис. 1.1.3. Создание нового проекта

В открывшемся окне необходимо ввести имя нового проекта, дополнительно можно указать его описание. Нажатие на кнопку «Создать» в указанном окне завершит процесс создания проекта. Новый проект будет открыт в главном окне студии.

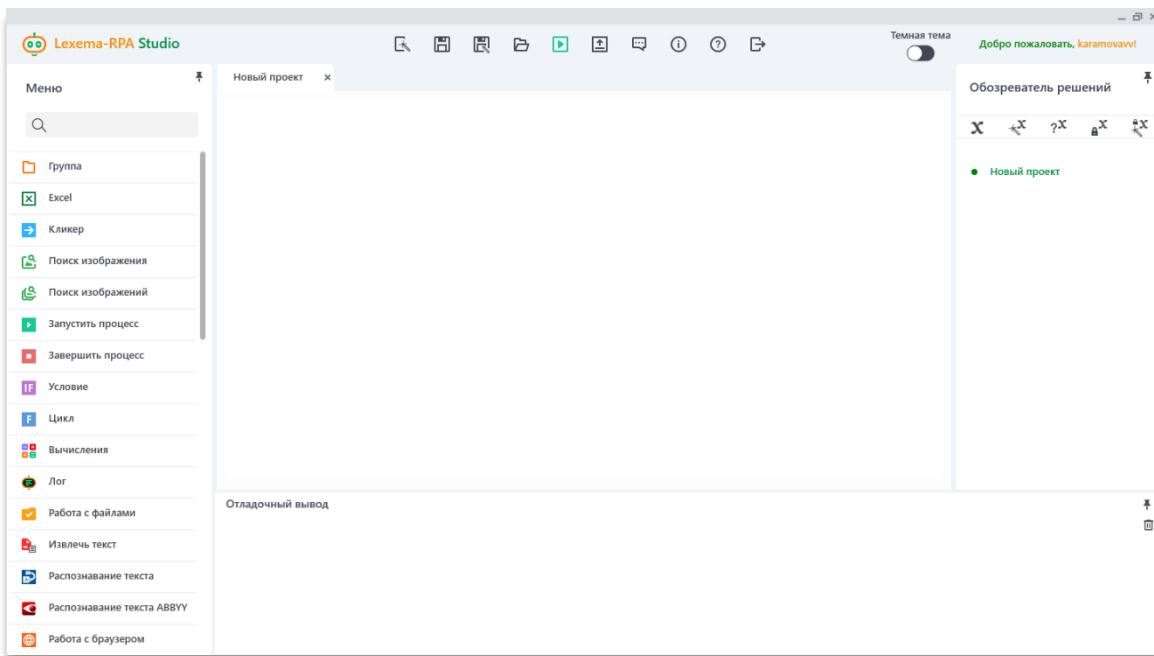


Рис. 1.1.4. Окно нового проекта

Сохранить

Кнопка «Сохранить» вызывает стандартный диалог сохранения файла.

Для нового проекта в диалоговом окне необходимо выбрать папку назначения и название файла, в котором будет сохранен проект (по умолчанию, предлагается название проекта, которое вы задали при его создании). Файлы проектов Lexema-RPA Studio имеют расширение «.гра». После выбора папки для сохранения, необходимо нажать на кнопку «Сохранить» (Рис. 1.1.5). По завершении операции, файл проекта будет сохранен по указанному расположению. Существующий проект будет сохранен в его текущем расположении.

Сохранить как

Кнопка «Сохранить как» вызывает стандартный диалог сохранения файла, аналогичный окну сохранения нового проекта.

В диалоговом окне предоставляется возможность для изменения наименования или расположения файла проекта.

Открыть

Кнопка «Открыть» вызывает стандартный диалог выбора файла.

Выбранный файл проекта будет загружен в главное окно студии.

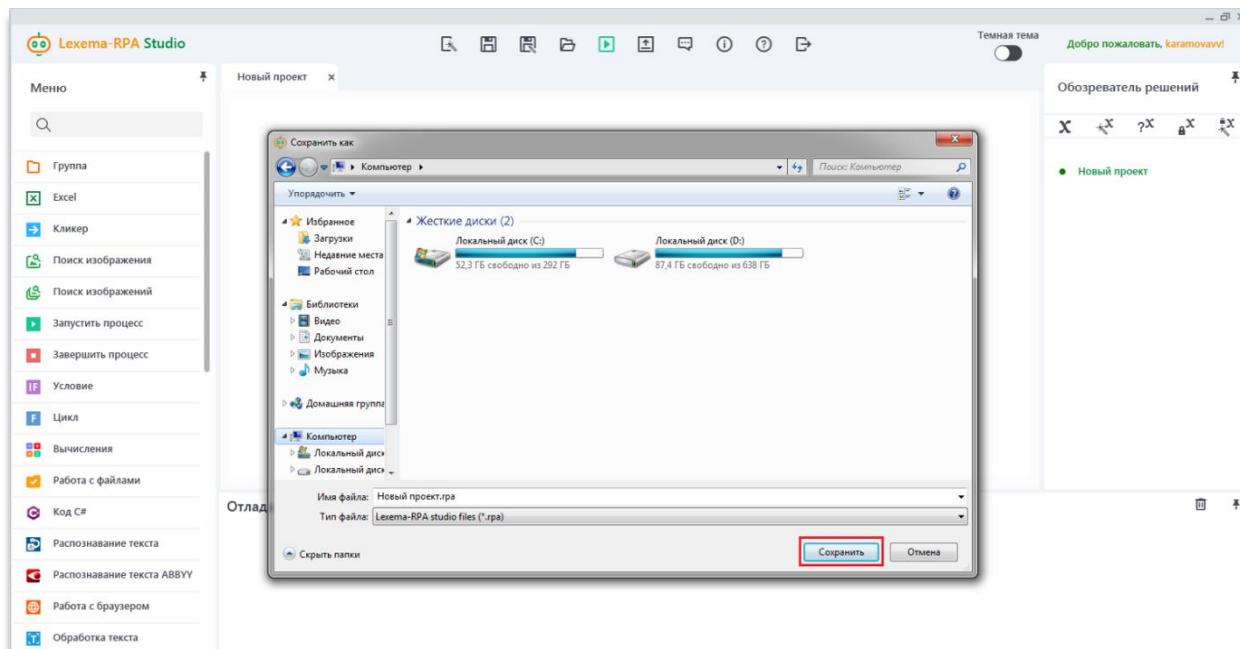


Рис. 1.1.5. Диалоговое окно сохранения проекта

Проиграть

Кнопка «Проиграть» запускает выполнение всех описанных в проекте действий.

Попытка запуска будет осуществляться только в случае, если в проекте имеется хотя бы одно действие. Действия выполняются в порядке, описанном в структуре проекта.

В случае возникновения ошибок или исключительных ситуаций, они будут выведены в область «Отладочный вывод» (далее будет дано подробное описание области отладочного вывода).

Опубликовать

Кнопка «Опубликовать»  вызывает диалог публикации текущего проекта в оркестратор роботов⁵.

Форум

Кнопка «Форум»  позволяет перейти на веб-сайт форума технической поддержки приложения - <http://talk.lexema.ru/>.

О программе

Кнопка «О программе»  выводит диалоговое окно, содержащее информацию о текущей версии студии.

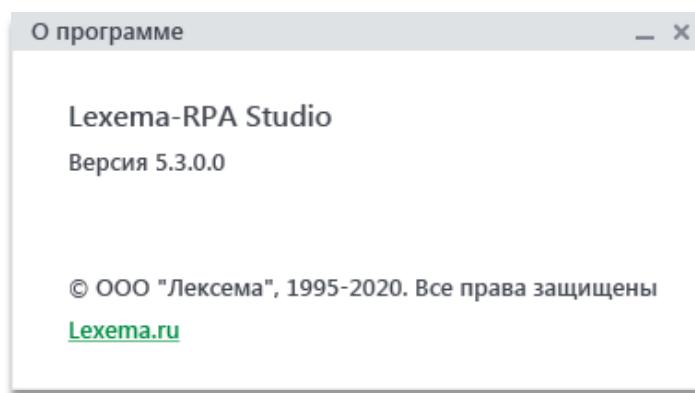


Рис. 1.1.6. Информация о текущей версии студии

Выйти

Кнопка «Выйти»  закрывает студию.

Меню

В разделе «Меню» выводится список доступных для добавления в проект модулей. Каждый модуль отвечает за выполнение конкретного действия или за взаимодействие с определенной программой.

Список доступных модулей

В программе Lexema-RPA Studio имеются следующие модули: Excel, Word, Кликер, Поиск изображения (2 варианта), Запустить/Завершить процесс, Условие, Цикл, Вычисления, Работа с файлами, Распознавание текста (обычное и ABBYY), SQL, Отправить/Считать/Переместить письмо, Break, JSON, XML, Интерфейс, Закрыть Excel-файл, While, Return, Try/Catch, Лог, Извлечь текст, PDF, HTTP OData, HTTP запрос, FTP запрос, Создать скриншот, Прокси.

⁵ Оркестратор роботов — сервис запуска роботов по составленному расписанию и событиям.

В списке доступных модулей выводятся все имеющиеся модули (рис. 1.1.7).

Поиск модуля

Поиск модуля можно выполнить с помощью текстового поля, расположенного над списком доступных модулей. Список будет фильтроваться по мере ввода поисковой фразы (рис. 1.1.8).

Добавление модуля в проект

Для добавления выбранного модуля в проект необходимо перетащить его в рабочую область с помощью указателя мыши. Новый модуль можно расположить в нужной позиции в общей последовательности действий. Для этого необходимо навести курсор на соседний с нужной позицией модуль. При этом соседний модуль будет подсвечен красным выделением и положение нового модуля будет отображаться в виде яркого выделения сверху или снизу: сверху – если новый модуль будет добавлен перед соседним, снизу – если после него (рис. 1.1.9 – 1.1.11).

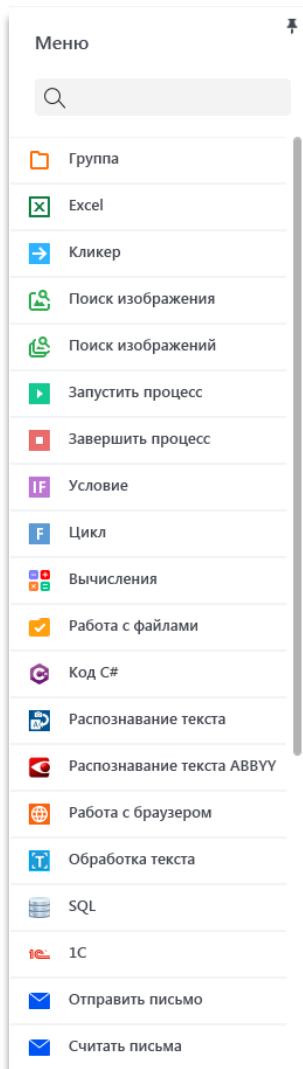


Рис. 1.1.7. Меню доступных модулей

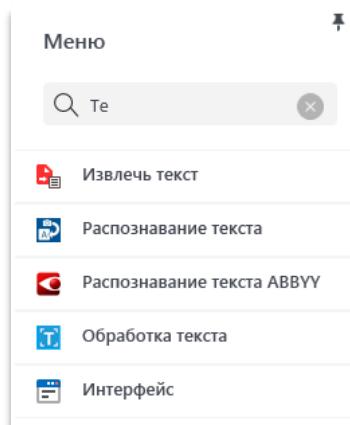


Рис. 1.1.8. Поиск в списке модулей

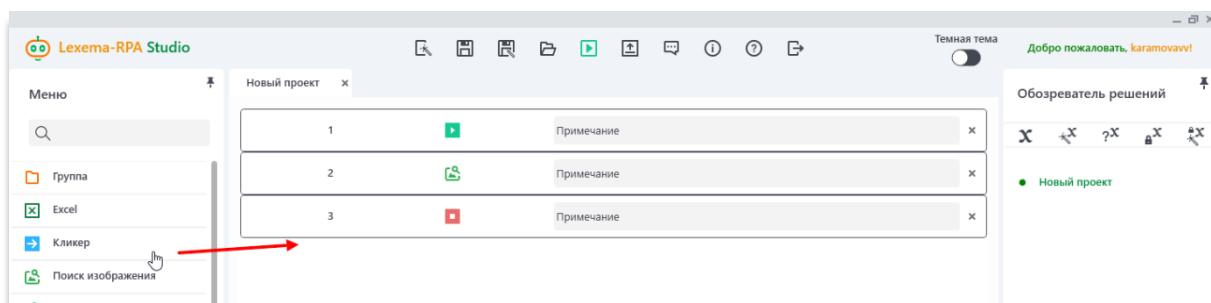


Рис. 1.1.9. Выбор модуля и перетягивание его в нужную область

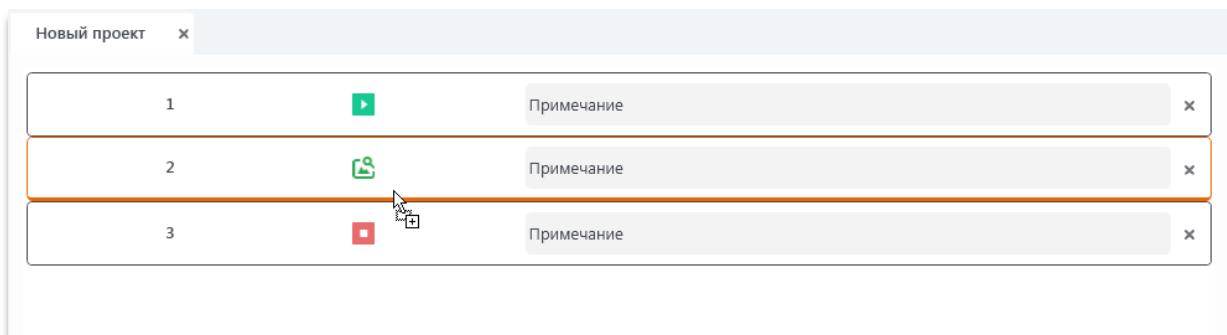


Рис. 1.1.10. Подсветка 2-го модуля при перемещении нового модуля после него

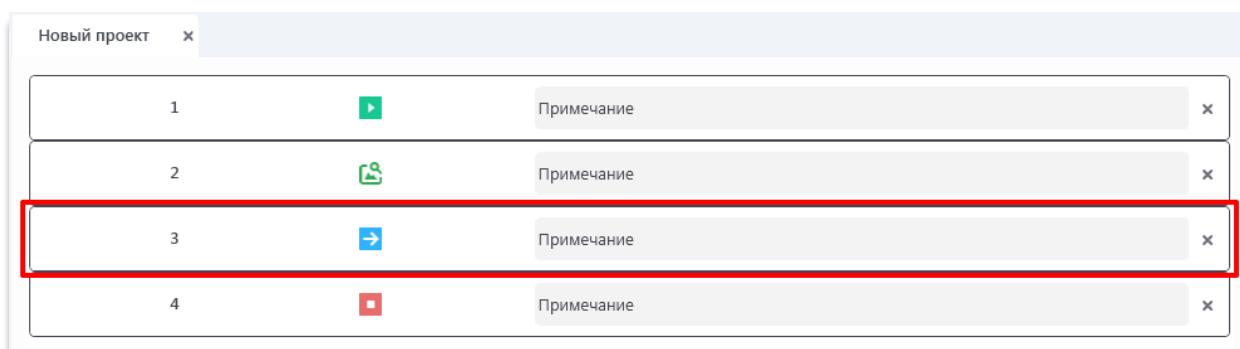


Рис. 1.1.11. Итоговое расположение модуля

Обозреватель решений

Раздел «Обозреватель решений» находится в правой части главного окна. В верхней части раздела расположена панель управления переменными и функциями, прямо под панелью - область текущих проектов. Активный проект (окно которого сейчас открыто) выделен зеленым цветом и помечен маркером зеленая точка.

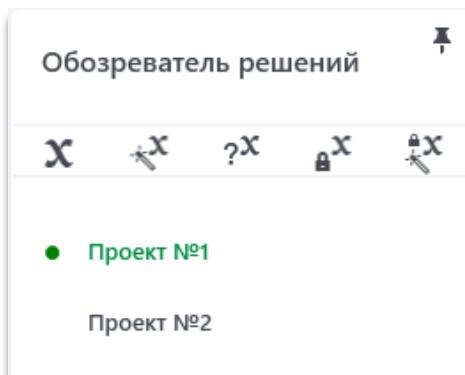


Рис. 1.1.12. Обозреватель решений

Панель управления переменными и функциями содержит следующие кнопки (перечисление слева направо): «Переменные», «Создать новую переменную», «Узнать значение переменной», «Локальное хранилище», «Создать локальную переменную», «Функции». При нажатии на одну из этих кнопок будут открыты или созданы те переменные или функции, которые относятся к активному проекту. Подробное описание кнопок будет приведено далее.

Обзор переменных

В рабочей области студии можно вывести информацию о доступных в активном проекте переменных.

Окно обзора переменных отображается во вкладке «Переменные» при нажатии на кнопку «Переменные»  в обозревателе решений и доступно только при наличии открытого проекта.

Переменные	
Название переменной	Значение
CurrentDirectory	D:\LexemaStudio
UserNameWindows	user
MachineName	LENOVO-G50

Рис. 1.1.13. Обзор переменных проекта

Окно содержит строку для поиска переменной по её названию и таблицу со списком всех доступных проекту переменных. Таблица имеет два столбца:

- Название переменной
- Значение переменной

По умолчанию, *переменные автоматически создаются при указании их в том или ином модуле*. При создании проекта автоматически создаются системные переменные:

- CurrentDirectory – директория, в которой установлена студия;
- UserNameWindows – название пользователя ОС;
- MachineName – имя компьютера.

При двойном щелчке по строке с переменной открывается диалоговое окно просмотра свойств переменной. В диалоговом окне предоставляется возможность перейти в окно редактирования свойств переменной, просмотреть дерево свойств переменной и удалить переменную из проекта. Подробное описание работы с переменными будет рассмотрено в главе 2.

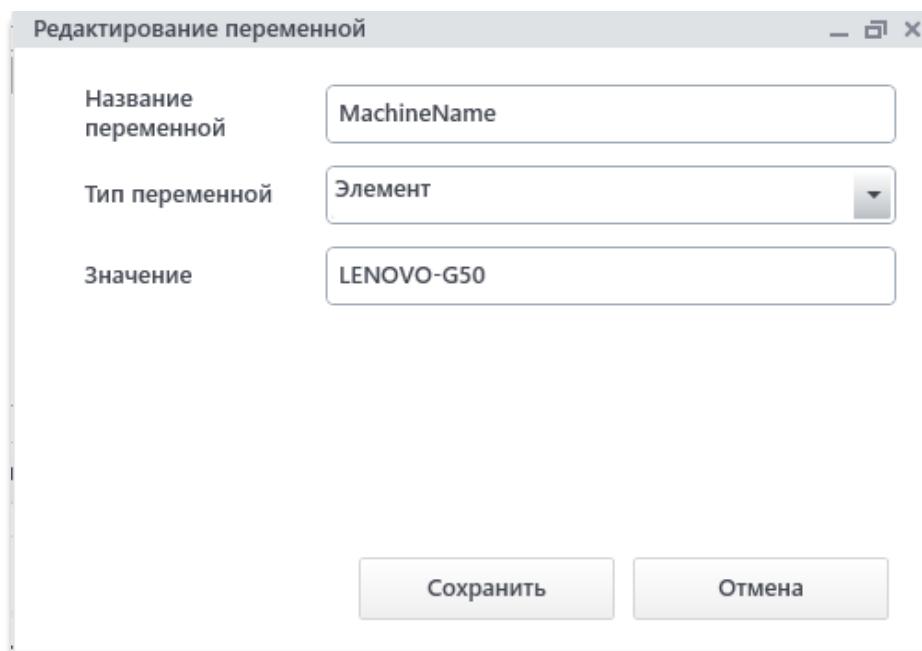


Рис. 1.1.14. Окно настройки переменной

Обзор локальных переменных

Lexema-RPA Studio позволяет хранить конфиденциальную информацию, например, пароли, логины или почтовые адреса, в переменных, которые будут зашифрованы и сохранены на вашем компьютере. Обзор ваших локальных переменных доступен по кнопке «Локальное хранилище»  в панели управления переменными и функциями. Окно состоит из трех столбцов:

- Имя – название переменной;
- Значение;
- Описание – ваш комментарий относительно переменной.

Локальное хра...		
Имя	Значение	Описание
email	rpa_support@lexema.ru	Почта тех-поддержки RPA

Рис. 1.1.15. Локальные переменные

При двойном нажатии по переменной открывается окно её редактирования, аналогичное окну создания локальной переменной.

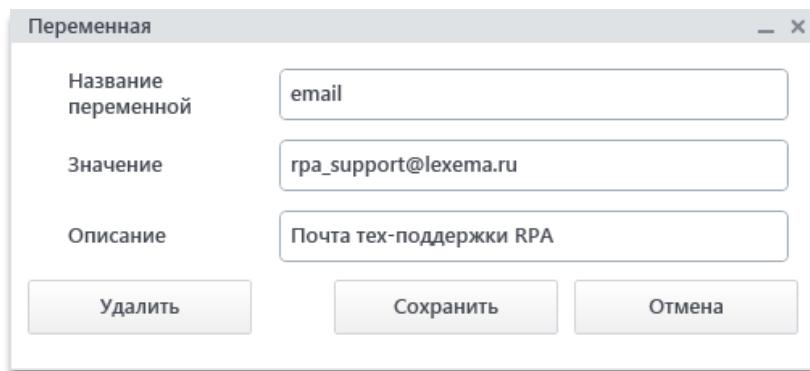


Рис. 1.1.16. Окно редактирования локальной переменной

Отладочный вывод

В раздел «Отладочный вывод» главного окна студии выводятся все сообщения, возникающие в ходе выполнения настроенных действий проекта после нажатия на кнопку «Проиграть». Сообщения выводятся для всех действий по порядку их выполнения в проекте.

Отладочный вывод
Проект "Проект №1 ". Робот отработал успешно. Начало работы робота - 11:41, время работы робота - 0,0528973с.
11:41:40.2161166 Модуль - Кликер (1), сообщение - Успешно!, время выполнения - 0,0109744с.
11:41:40.2330745 Модуль - Excel (2), сообщение - Успешно!, время выполнения - 0,0419229с.

Рис. 1.1.17. Пример успешного вывода

В отладочном выводе выводится информация о времени начала работы, общее время работы. Для каждого модуля выводится информация о начале времени его выполнения и затраченное время.

В случае возникновения исключения в работе модуля, робот приостановит выполнение, и в отладочном выводе, в последней строке, будет содержаться информация об ошибке. Текст последней строки отладочного вывода будет написан красным цветом.

Отладочный вывод

```
Проект "Проект №1 ". Робот завершился с ошибкой. Начало работы робота - 11:43, время работы робота - 2,0309655с.  
11:43:26.1958633 Модуль - Кликер (1), сообщение - Успешно!, время выполнения - 0,0199463с.  
11:43:26.2237891 Модуль - Excel (2), сообщение - Успешно!, время выполнения - 0,0219397с.  
11:43:26.2518555 Модуль - Telegram (3), сообщение - Значение не может быть неопределенным.
```

Рис. 1.1.18. Пример вывода при исключительной ситуации

Глава 2. Основы работы

Теперь, когда вы познакомились с интерфейсом студии, пришло время выяснить, как работать с этой программой. В этой главе вы узнаете о переменных и функциях, а также освоите маленькие хитрости (горячие клавиши), которыми сможете воспользоваться для быстрой работы.

Переменные

Переменные предназначены для хранения данных, которые должны быть доступны для всех действий проекта. Большинство модулей поддерживает чтение и запись данных в переменные.

К наименованию переменных предъявляется следующее требование: **название переменной должно всегда начинаться с символов ‘v.’**.

Обращаться к переменным можно в полях окон настройки модулей приложения. При обращении необходимо указывать полное название переменной согласно описанному выше требованию.

В проекте можно создавать переменные четырех типов:

- Элемент – переменная, имеющая одно значение. Это может быть строка или число;
- Список – переменная, состоящая из нескольких элементов;
- Таблица – переменная в виде таблицы, в ячейках которой содержатся элементы;
- Объект – переменная, имеющая именованные поля - ключи и их значения-элементы.

Для создания переменной нажмите на кнопку «Создать переменную» в обозревателе решений, предварительно выбрав нужный проект (сделав его активным). Для создания локально хранимой и зашифрованной переменной необходимо нажать кнопку «Создать локальную переменную». После выполнения действия откроется окно создания переменной. Локальные переменные могут создаваться только с типом «Элемент», поэтому окно их настройки немного отличается от настройки обычных переменных.

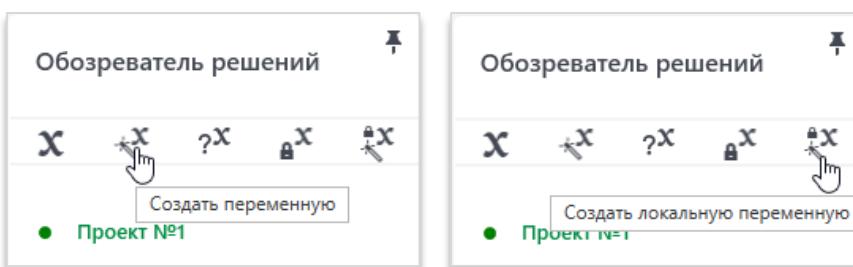


Рис. 1.2.1. Кнопки создания переменной

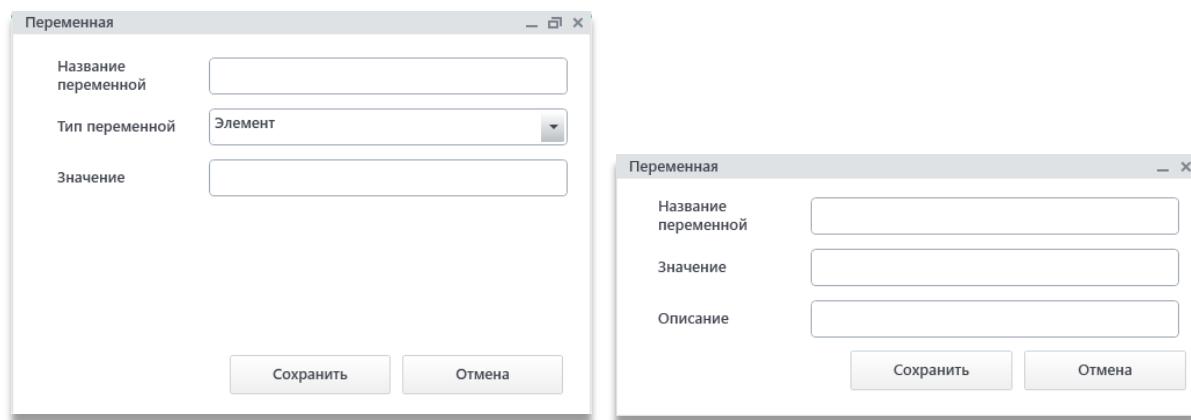


Рис. 1.2.2. Окна создания переменной (слева – обычной, справа – локальной)

В окне создания переменной укажите название переменной согласно требованиям приложения (**в начале наименования обязательно должны быть символы v.**) и выберите тип переменной из списка. При изменении типа переменной изменяются доступные настройки.

Для переменных с типом «Элемент» доступны следующие настройки:

- поле для ввода значения.

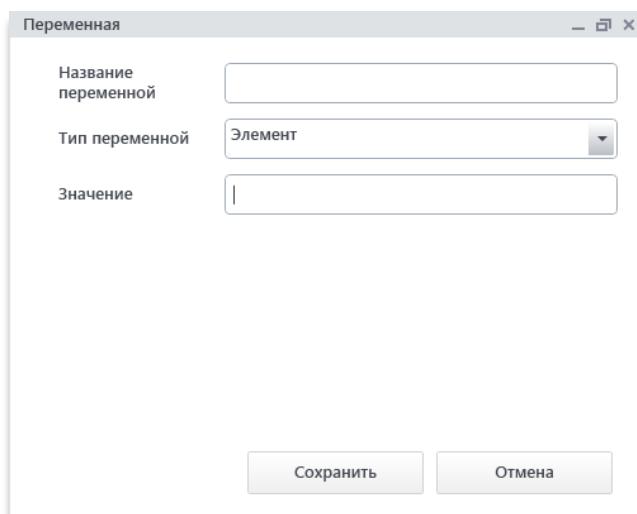


Рис. 1.2.3. Окно создания обычной переменной-элемента

Переменные с типом «Список» хранят набор (массив) переменных. Обращение к каждому элементу списка осуществляется по его индексу, индекс начинается с 0. Индекс указывается в квадратных скобках, следующих сразу за названием переменной. Например, для обращения ко второму элементу – `v.array[1]`.

Заполнение переменных-списков производится посредством ввода значения в поле «Значение», затем нажатием кнопки «Добавить», после чего введенный элемент добавится в список элементов в нижней части окна.

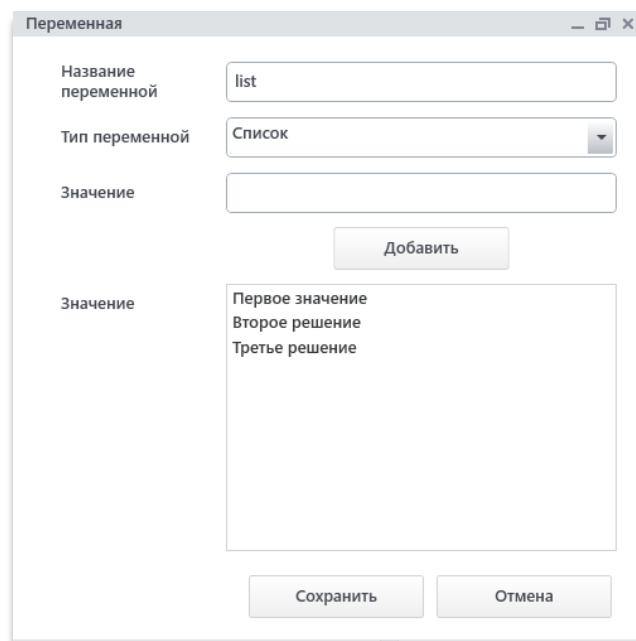


Рис. 1.2.4. Создание переменной-списка

Переменные с типом «Таблица» представляют собой таблицу (двумерный массив), т.е. это список, элементами которого являются списки. Другими словами, это набор простых элементов, доступ к которым осуществляется по двум индексам. Наглядно двумерный массив удобно представлять в виде таблицы, в которой n строк и m столбцов, а под ячейкой таблицы, стоящей в i -й строке и j -м столбце, понимают некоторый элемент массива $a[i-1, j-1]$ (так как индекс начинается с нуля). Это означает, что для обращения к элементу в 3 строке и 5 столбце, отвечающему за ячейку таблицы, необходимо указывать два индекса: `v.table[2, 4]`.

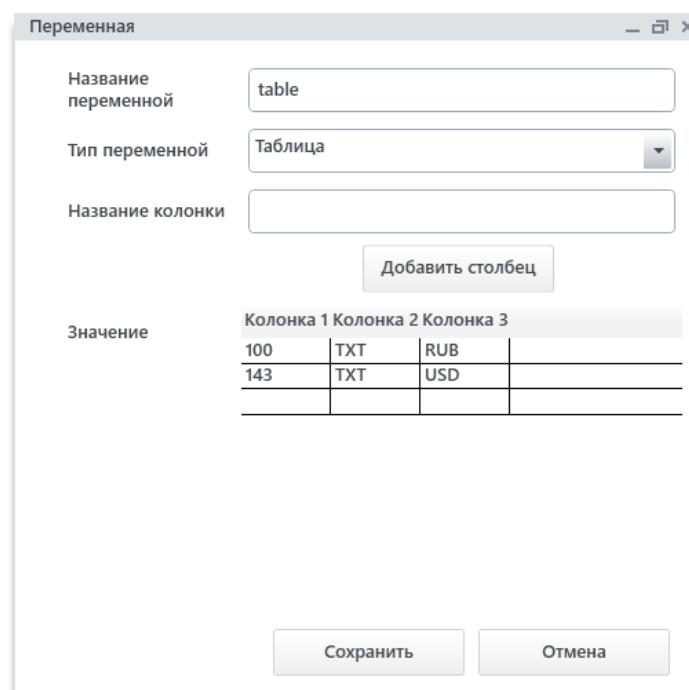


Рис. 1.2.5. Создание переменной-таблицы

Для заполнения значений переменных с типом «Таблица» становится доступной таблица, набор столбцов которой можно расширять с помощью кнопки «Добавить столбец». Добавление новых строк производится по нажатию клавиши «Enter». Вводить значения можно для каждой ячейки. **Незаполненные столбцы не будут использоваться при создании переменной.**

И последний возможный тип переменных в программе – переменные-объекты. Для создания переменной необходимо заполнять поля «Ключ» и «Значение» указанного ключа. В примере показана переменная `v.person`, которая состоит из двух полей – `name` и `age`, значения которых `John` и `18` соответственно. Для обращения к полям такой переменной в программе необходимо указать название переменной с символом `«v.»` и через точку поле, к которому хотите обратиться: `v.person.name` (результатом будет строка `John`).

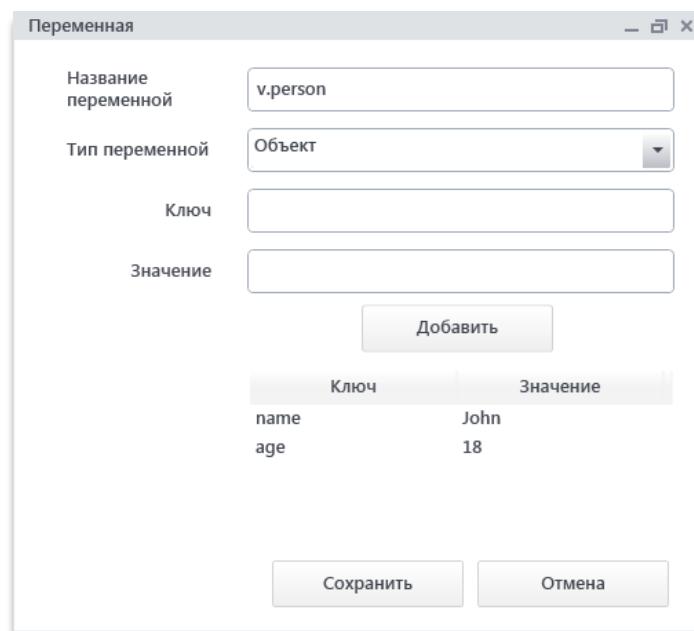


Рис. 1.2.6. Создание объектной переменной

При нажатии в обозревателе решений на кнопку «Узнать значение переменной» открывается окно, с помощью которого можно проверить значение переменной активного проекта (рис. 1.2.7 – 1.2.8). **Для проверки значения необходимо ввести полное наименование переменной (включая символы `v.`).**

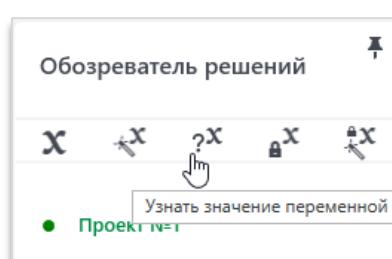


Рис. 1.2.7. Кнопка «Узнать значение переменной»

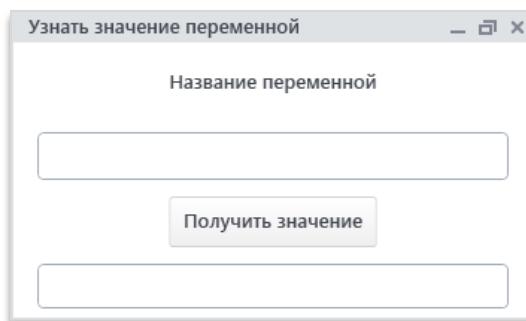


Рис. 1.2.8. Окно проверки значения переменной

Функции

Предустановленный набор вспомогательных функций позволяет выполнять различные вычисления в действиях проекта.

Для вызова встроенной функции необходимо перед её назвианием ввести знак #, например, `#GetDate()`. Аргументы функции, над которыми должна производиться операция, перечисляются в круглых скобках через запятую (если функция работает с несколькими аргументами). Между наименованием вызываемой функции и скобками с её аргументами не должно быть пробела.

Рабочая область

Это основной раздел главного окна студии. Здесь осуществляется настройка структуры проекта и выполняемых в нём действий.

Структура проекта

Действия добавляются с помощью меню доступных модулей. После добавления модуля в структуру проекта он отображается в виде элемента, обозначающего выполняемое действие.

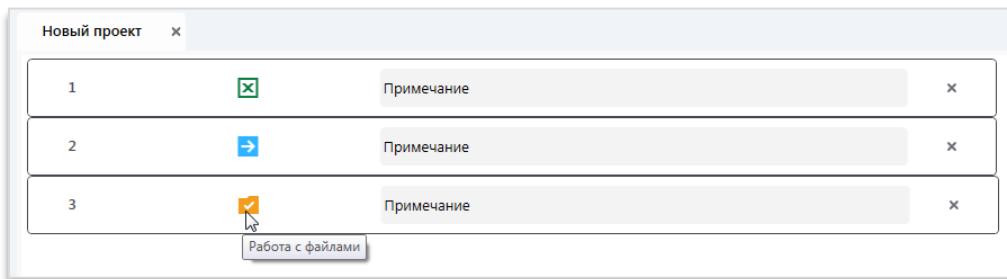


Рис. 1.2.9. Структура проекта

Элемент действия имеет несколько составляющих:

- порядковый номер действия;
- название действия;

- настраиваемое пользователем примечание;
- кнопку удаления действия.

В структуре проекта по нажатию правой кнопкой мыши на добавленное действие доступно контекстное меню со следующими пунктами:

- отключить/включить;
- снять/поставить точку остановки;
- запустить с действия;
- копировать;
- вставить;
- удалить.

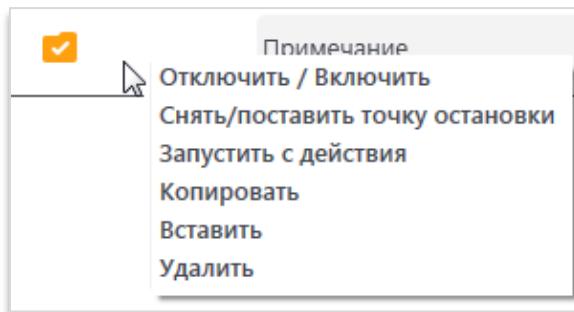


Рис. 1.2.10. Конекстное меню структуры проекта

Пункт «Отключить/Включить» позволяет переключать активность выбранного действия. Выключенное действие будет пропущено при запуске проекта.

Пункт «Снять/поставить точку остановки» создаёт или отключает точку остановки выполнения проекта. Необходимо для отладки проекта, при установленной точке выполнение робота приостановится перед отмеченым модулем.

Пункт «Запустить с действия» позволяет запустить робота с выбранного действия, то есть все ранее стоящие модули будут пропущены.

Пункт «Копировать» копирует в буфер обмена выбранное действие для последующей вставки.

Пункт «Вставить» вставляет скопированное действие **после выделенного действия**. Данный пункт доступен также при вызове контекстного меню в рабочей области главного окна студии.

Пункт «Удалить» позволяет удалить выбранный модуль.

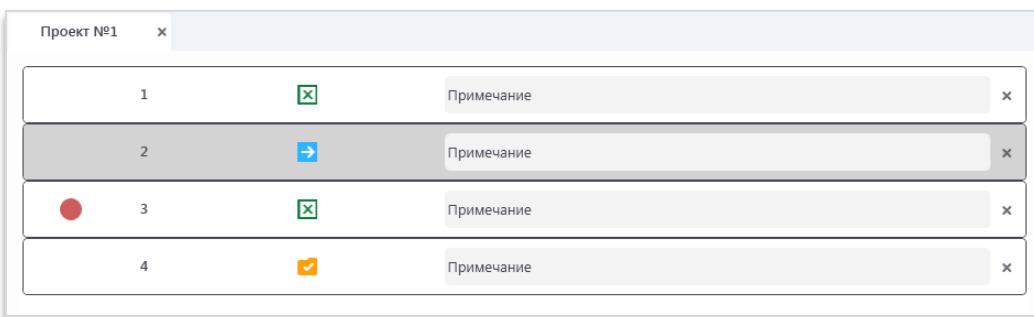
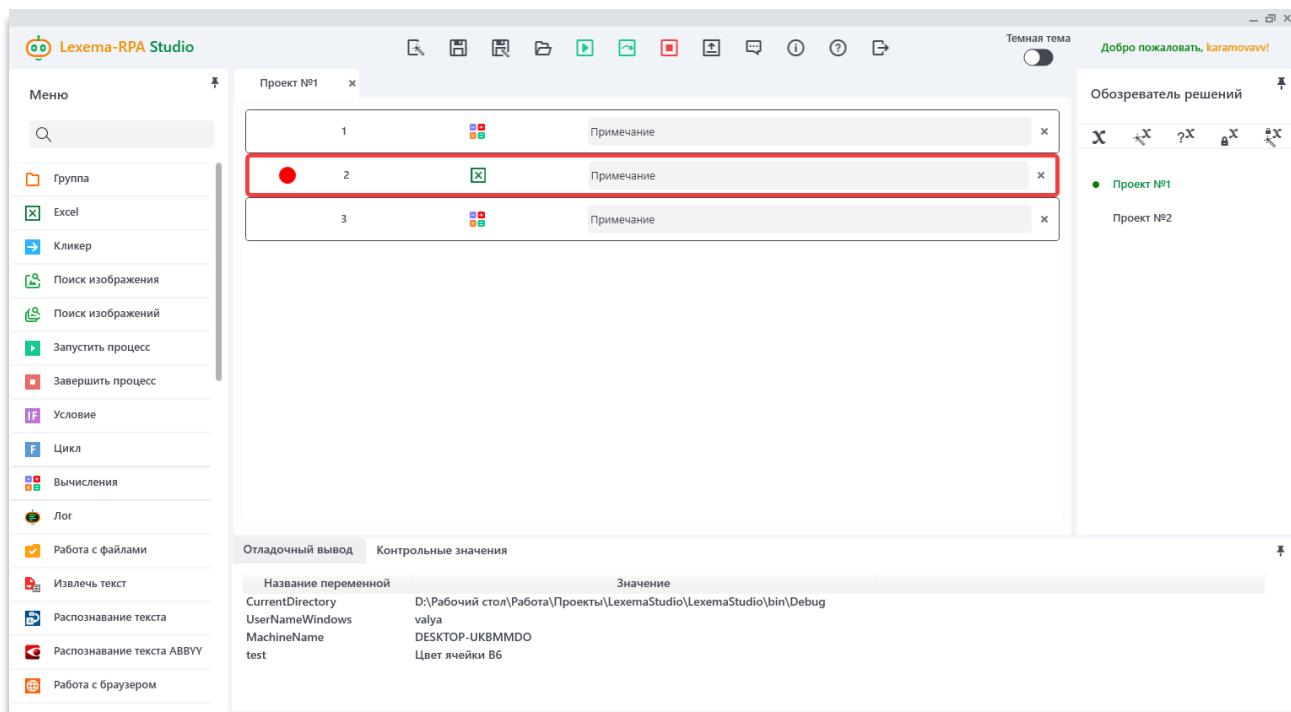


Рис. 1.2.11. Проект с отключенным действием и точкой остановки

Точка остановки

Точка остановки – это инструмент, который помогает отлаживаться при написании робота. Такой инструмент позволяет остановить работу робота перед выполнением выбранного модуля, при этом в панели отладочного вывода отображаются текущие переменные и их значения, модуль, который должен будет выполняться после продолжения подсвечивается красной рамкой, а в панели инструментов появляются две новые кнопки, позволяющие управлять *пошаговой отладкой*.



Пошаговая отладка позволяет выполнять каждый модуль робота друг за другом, при этом можно следить за значениями всех переменных. Для выполнения подсвечиваемого модуля и остановки робота после его выполнения, то есть, для перехода к следующему модулю (шагу), необходимо нажать кнопку «Следующий шаг отладки»  . Для прекращения выполнения работы робота необходимо нажать кнопку «Остановить»  . Если требуется остановить робота в другом месте, через несколько модулей, то во время отладки можно поставить новую

точку остановки на модуле, перед выполнением которого необходимо остановить работу робота, и нажать «Продолжить выполнение» . Изменения, внесенные в модули во время отладки, в текущем выполнении *не будут*.

Настройка добавленных модулей

Окно настройки свойств модуля, добавленного в структуру проекта, открывается с помощью двойного щелчка левой кнопкой мыши по указанному действию. Набор свойств, доступных в этом окне, зависит от выбранного модуля. Подробнее о доступных свойствах каждого модуля будет описано далее в руководстве.

Горячие клавиши

Приложение поддерживает следующий набор комбинаций горячих клавиш:

Комбинация клавиш	Действие
<i>Ctrl+N</i>	Создать новый проект. Открывает диалог создания нового проекта.
<i>Ctrl+S</i>	Сохранить проект. Сохраняет проект в текущем расположении.
<i>Ctrl+Shift+S</i>	Сохранить проект как. Аналогично кнопке «Сохранить как» открывает диалог для указания нового расположения или имени файла проекта.
<i>Ctrl+O</i>	Открыть проект. Открывает сохраненный ранее проект с помощью системного диалога выбора файлов.
<i>Esc</i>	Закрыть программу.
<i>F5</i>	Запустить выполнение робота в текущем проекте.
<i>F1</i>	Перейти в справку по адресу http://wikie.lexema.ru/ .
<i>F5</i>	Запустить робота
<i>F10</i>	Перейти к следующему шагу отладки

ЧАСТЬ II. МОДУЛЬ «EXCEL»

Глава 3. Основные моменты модуля

Модуль «Excel» является самым обширным модулем в программе. Данный модуль предназначен для обработки Excel-файлов в фоновом режиме, незаметном для пользователя компьютера. Модуль последовательно выполнит заданный ему список команд.

Интерфейс модуля

Окно модуля состоит из нескольких разделов: «Общие настройки» (верхняя часть окна), «Добавить команду», «Список команд» и окна с предварительным просмотром файла.

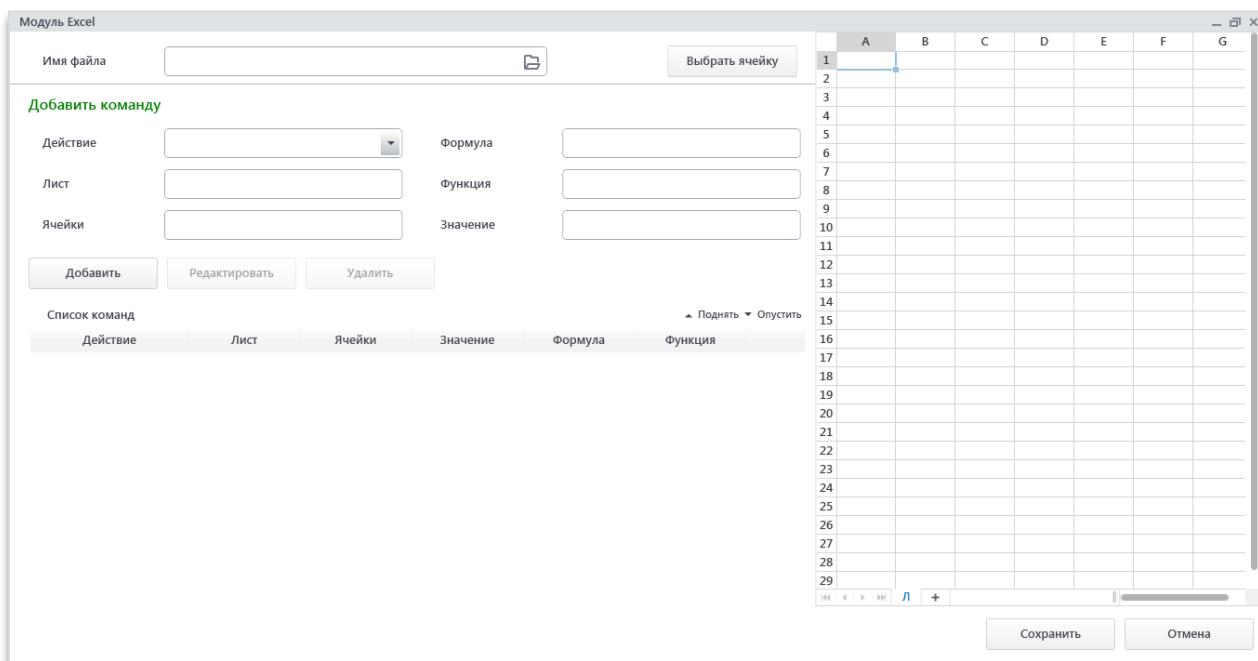


Рис. 2.3.1. Окно модуля Excel

Окно предварительного просмотра позволяет просматривать все листы загруженного файла с помощью вкладок с названиями листов и полосы прокрутки. Также есть возможность вписывать в ячейки свои значения, применять различное форматирование, но эти изменения сохранены не будут, то есть данный функционал доступен только для «примерки».

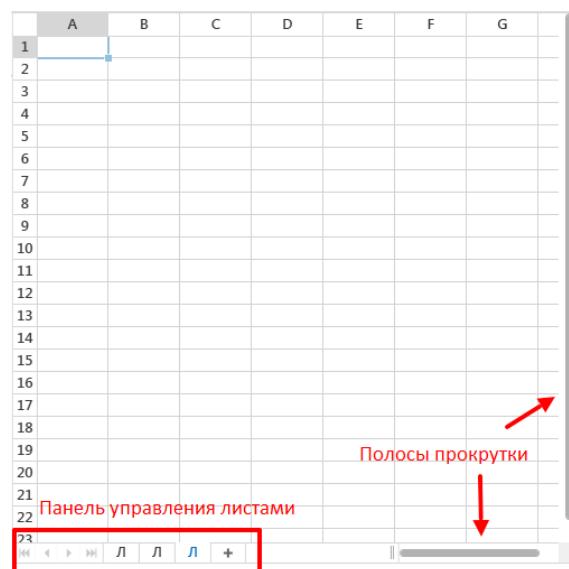


Рис. 2.3.2. Элементы управления окна предварительного просмотра

Раздел «Общие настройки»

Загрузить файл

Раздел «Общие настройки» состоит из поля «Имя файла» и кнопки «Выбрать ячейку».

Работу с модулем Excel советуем начинать с выбора рабочего файла. После того, как по кнопке «Открыть файл» в поле «Имя файла» будет выбран файл, необходимо подождать несколько секунд и указанный файл будет отображен в окне предварительного просмотра (рис. 2.3.3).

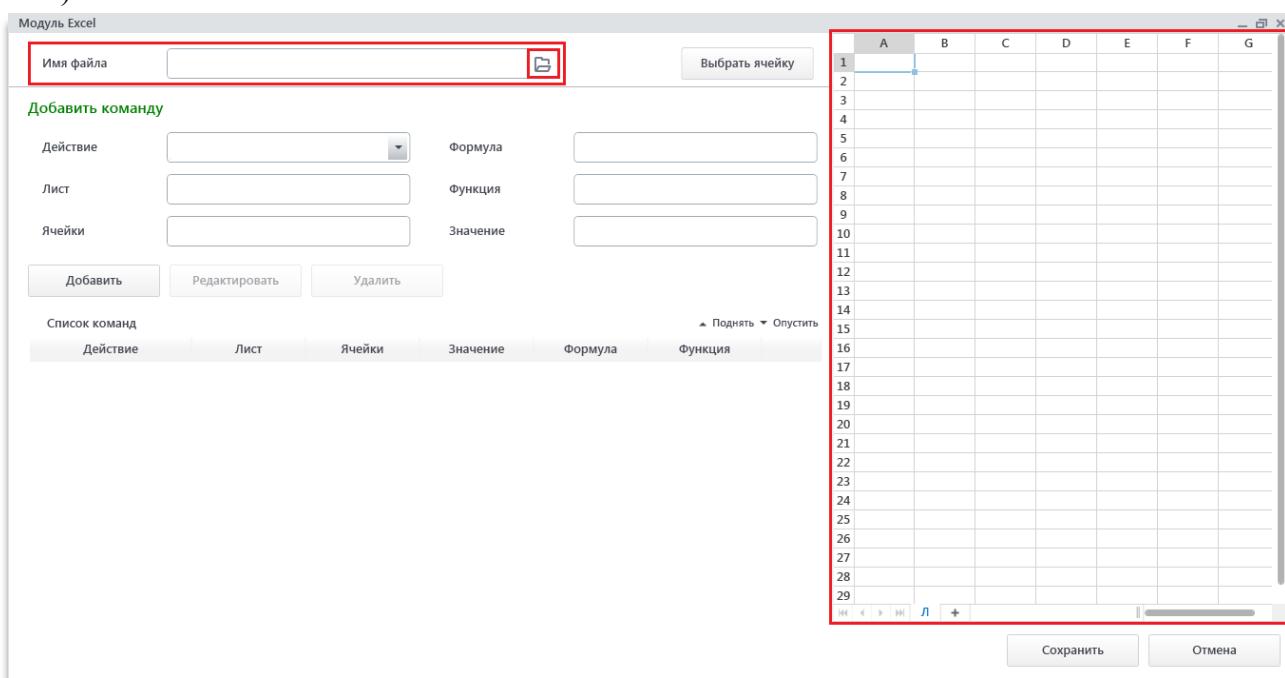


Рис. 2.3.3. Общие настройки

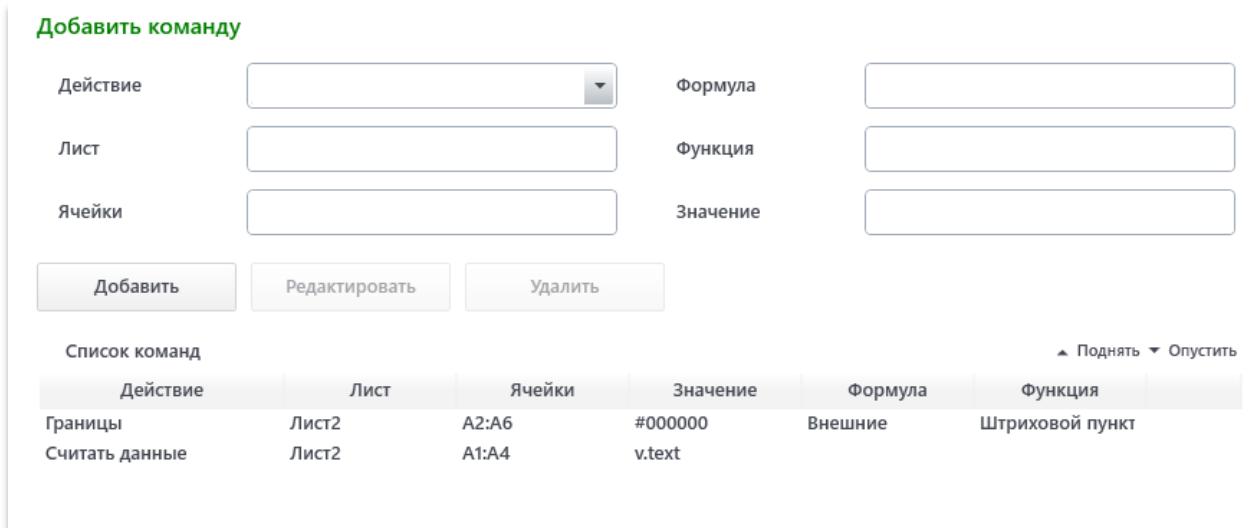
Разделы «Добавить команду» и «Список команд»

Раздел «Добавить команду» является основным «полотном» данного модуля. В нем создаются действия, имитирующие работу пользователя, например, такие как считать некоторый диапазон ячеек, затем перенести его на другой лист, удалить дубликаты и построить круговую диаграмму. В данном случае было описано 4 действия. Эти 4 действия необходимо последовательно выбирать и добавлять в раздел «Список команд» - таблицу, отображающую все действия, которые будут выполняться данным модулем. Набор полей в данном разделе не всегда одинаков – он меняется в зависимости от выбранного действия в первом поле.

Кнопки «Добавить», «Редактировать» и «Удалить» предназначены для работы с разделами «Добавить команду» и «Список команд».

Добавление команды

Для добавления команды предусмотрена кнопка «Добавить». Она заносит созданное действие в таблицу «Список команд».



Действие	Лист	Ячейки	Значение	Формула	Функция
Границы	Лист2	A2:A6	#000000	Внешние	Штриховой пункт
Считать данные	Лист2	A1:A4	v.text		

Рис. 2.3.4. Добавление команды

При выборе строки в списке команд информация по выбранному действию отобразится в разделе «Добавить команду».

Помимо описанного функционала, кнопка «Добавить» служит для копирования уже имеющихся действий в списке команд. Для этого необходимо выбрать строку в списке команд и нажать на кнопку «Добавить». Таким образом, если нужно выполнить действие несколько раз, достаточно его копировать, а не создавать вновь.

Добавить команду

Действие	Границы	Расположение	Внешние																		
Лист	Лист2	Стиль	Штриховой пунктир																		
Ячейки	A2:A6	Цвет	#000000 <input type="color"/>																		
<input type="button" value="Добавить"/> <input type="button" value="Редактировать"/> <input type="button" value="Удалить"/>																					
Список команд ▲ Поднять ▼ Опустить <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Действие</th> <th>Лист</th> <th>Ячейки</th> <th>Значение</th> <th>Формула</th> <th>Функция</th> </tr> </thead> <tbody> <tr> <td>Границы</td> <td>Лист2</td> <td>A2:A6</td> <td>#000000</td> <td>Внешние</td> <td>Штриховой пункт</td> </tr> <tr> <td>Считать данные</td> <td>Лист2</td> <td>A1:A4</td> <td>v.text</td> <td></td> <td></td> </tr> </tbody> </table>				Действие	Лист	Ячейки	Значение	Формула	Функция	Границы	Лист2	A2:A6	#000000	Внешние	Штриховой пункт	Считать данные	Лист2	A1:A4	v.text		
Действие	Лист	Ячейки	Значение	Формула	Функция																
Границы	Лист2	A2:A6	#000000	Внешние	Штриховой пункт																
Считать данные	Лист2	A1:A4	v.text																		

Рис. 2.3.5. Отображение команды

Редактирование команды

Для редактирования команды предусмотрена кнопка «Редактировать». Чтобы изменить команду, выделите ее в списке команд, внесите необходимые изменения в разделе «Добавить команду» и нажмите на кнопку «Редактировать» для сохранения изменений.

Изменение порядка действий

Для изменения порядка действий предусмотрены кнопки «Поднять» и «Опустить». Кнопки расположены справа сверху от списка команд.

Изменение порядка действий бывает необходимо, когда требуется добавить действие, которое не предусмотрели сразу и не внесли в список команд. В таком случае вы можете добавить действие стандартным способом, а затем переместить его.

Список команд ▲ Поднять ▼ Опустить					
Действие	Лист	Ячейки	Значение	Формула	Функция
Границы	Лист2	A2:A6	#000000	Внешние	Штриховой пункт
Считать данные	Лист2	A1:A4	v.text		
Добавить строку	Лист2	12			

Список команд ▲ Поднять ▼ Опустить					
Действие	Лист	Ячейки	Значение	Формула	Функция
Границы	Лист2	A2:A6	#000000	Внешние	Штриховой пункт
Добавить строку	Лист2	12			
Считать данные	Лист2	A1:A4	v.text		

Рис. 2.3.6. Изменение порядка действий

Удаление команды

Кнопка «Удалить» удаляет выбранную строку из списка команд.

Отключение команды

Если вы не хотите удалять команду, так как хотели бы, например, проверить, как работает робот без неё, а затем вернуть её в работу, то можно отключить команду, нажав на соответствующую ей строку правой кнопкой мыши и отключить или, наоборот, включить её.

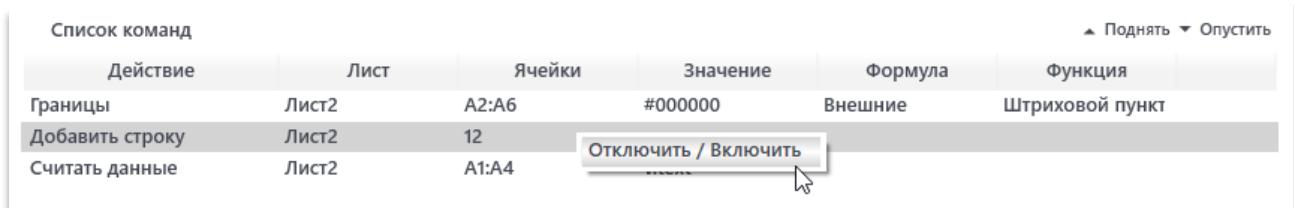


Рис. 2.3.7. Контекстное меню команды

Выбор ячейки

Кнопка «Выбрать ячейку» работает для каждого действия по-своему, но смысл у неё один – она заполняет поле «Ячейки» или «Диапазон» (в зависимости от того, какое из них используется в выбранном действии) названием выбранной в окне предварительного просмотра ячейки / диапазона и вписывает в поле «Лист» название листа, на котором находятся выбранные ячейки. **В некоторых действиях по кнопке заполняется только одна ячейка, хотя был выбран диапазон – это означает, что в данном действии использование диапазона невозможно.**

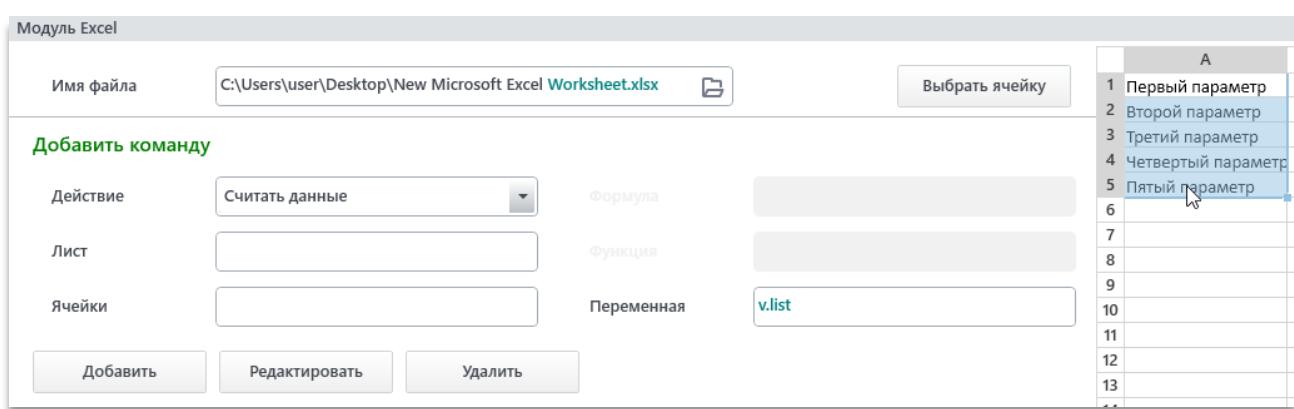


Рис. 2.3.8. Выбор диапазона в окне предварительного просмотра

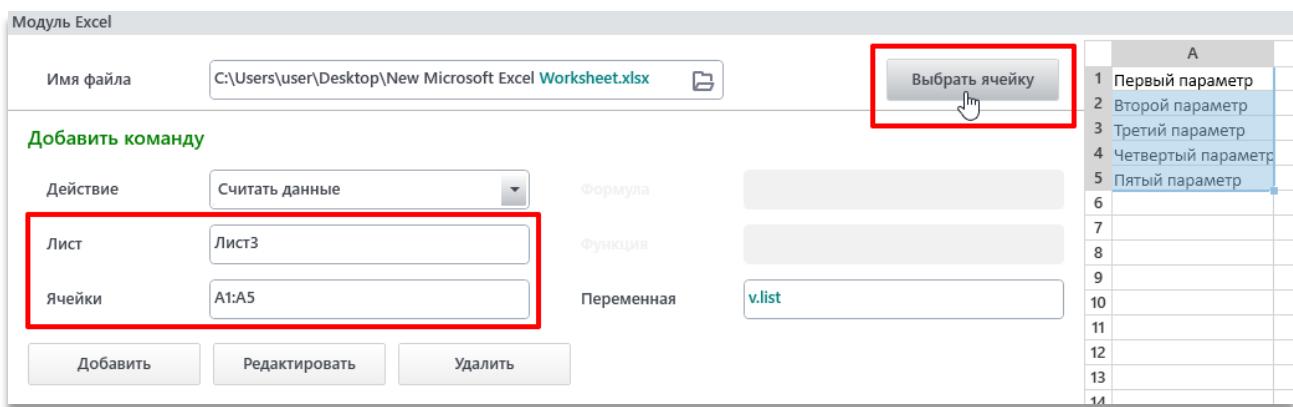


Рис. 2.3.9. Выбрать ячейку

В некоторых действиях кнопка «Выбрать ячейку» заполняет номера колонок, строк, какие-либо значения. При работе с файлами рекомендуется её использование для минимизации возникновения синтаксических ошибок.

Сохранение команды

Для сохранения команды в модуле Excel нажмите на кнопку «Сохранить» в правом нижнем углу модуля.

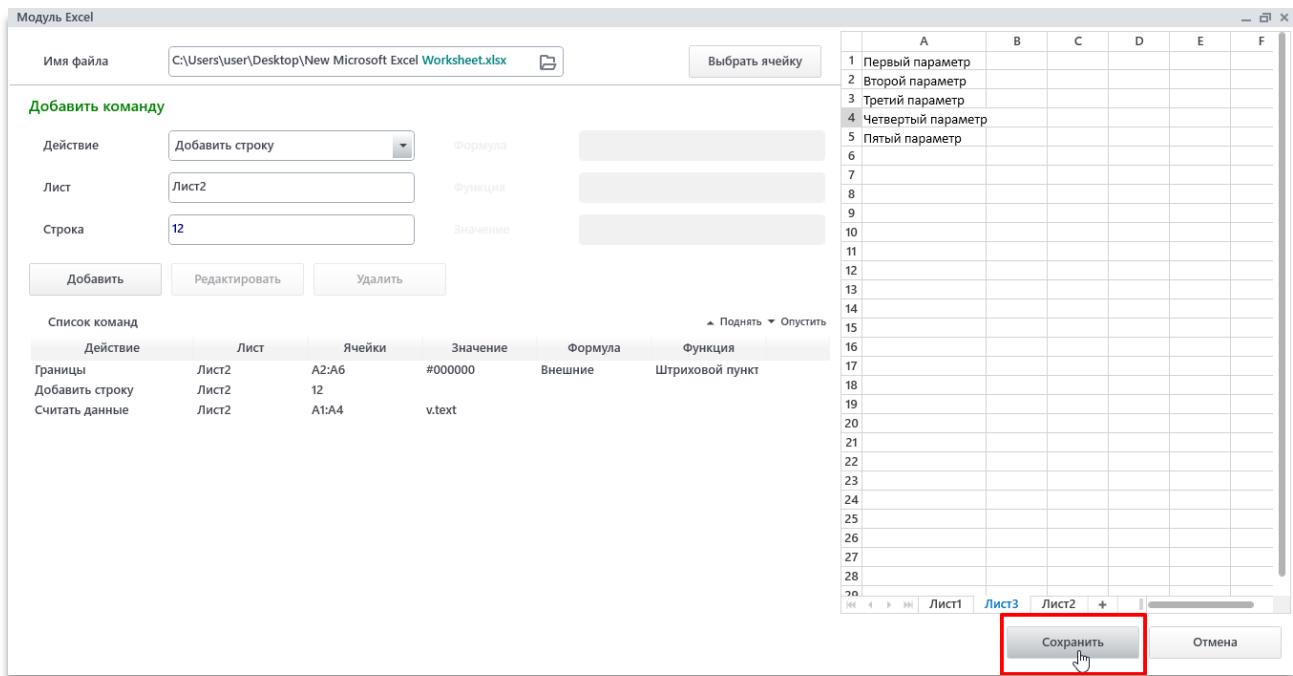


Рис. 2.3.10 Сохранение в модуле Excel

Глава 4. Набор полей в составе модуля

Для каждого действия в разделе «Добавить команду» существует свой набор необходимых для него полей, который будет отображен при выборе действия. По умолчанию, отображены поля «Лист», «Ячейки», «Формула» и «Переменная». Почти все команды имеют именно такой набор полей, но некоторые требуют несколько иной информации для выполнения действия.

Перечень возможных полей с правилами их заполнения

1. Лист – поле, в которое необходимо указать имя или номер листа (начиная с 0), на котором необходимо выполнить указанное действие;
2. Ячейки – поле, в котором ожидается либо адрес одной ячейки (A1), либо диапазон ячеек (A1:C10). Если адрес ячейки вычисляется динамически и неизвестен, то можно использовать номера колонок и строк ячеек, то есть вместо «C1» можно написать «3,1», где 3 – номер колонки, 1 – номер строки. Так же можно указать диапазон – вместо «A3:D4» - «1,3:4,4». Второй формат написания ячеек может быть использован только по отношению к полям «Ячейка(-и)», но невозможен в полях «Формула» (см. след. пункт);
3. Формула – поле для ввода формулы в стандартном формате Excel. Например «=ЕСЛИ(A2 > B2:1:0)». Для более удобного ввода длинных формул поле можно расширить, нажав на кнопку со стрелочкой в поле ввода:



Рис. 2.4.1. Поле «Формула»

4. Переменная – поле для ввода названия переменной, в которую будет помещен результат выполнения действия. Название вводится с символов «v.»;
5. Значение – поле для ввода текста или названия переменной, в которой лежит текст, используемый в выбранном действии;
6. Столбец – номер (с 1) или название столбца;
7. Стока – номер строки (с 1). В некоторых действиях («Найти данные», «Найти лист») – текстовая строка для поиска в файле;
8. Путь к файлу – путь до папки, в которую будет помещен файл. Можно выбрать с помощью диалога выбора папки по нажатию на кнопку в поле:



Рис. 2.4.2. Поле «Путь к файлу»

9. Имя файла – название файла, можно указать как с расширением, так и без, в таком случае будет создан файл с расширением `xlsx`;
10. Путь копируемого файла – путь до файла вместе с его расширением, который будет использоваться в данном действии помимо основного файла для копирования данных из него. Можно выбрать с помощью диалога выбора файлов по нажатию на кнопку в поле:



Рис. 2.4.3. Поле «Путь копируемого файла»

11. Лист копируемого файла – название или номер листа (с 0) в копируемом файле;
12. Колонки (ч/з ;) – аналогично полю «Столбец», только здесь возможно их перечисление через символ «;»;
13. Значения (ч/з ;) – строковые значения (или переменные, хранящие их), перечисленные через «;»;
14. Стока, колонка – номер строки и номер колонки, перечисленные через запятую;
15. Ячейка из диапазона значений – любая ячейка из таблицы, к которой применяется действие (например, если таблица A1:D10, то любая ячейка из этого диапазона – B5);
16. Функция – выпадающий список с набором функций, который меняется в зависимости от выбранного действия; обязательен выбор одного из представленных значений;
17. По убыванию – флажок, использующийся в действии «Добавить сортировку», который ставится, если необходимо сортировать по убыванию, и не ставится в противном случае;
18. Диапазон таблицы – аналогично полю «Ячейки», но с обязательным условием, что это должен быть именно диапазон (то есть должен присутствовать символ «:»);
19. Лист для вставки – название или номер листа (с 0);
20. Уникальные столбцы – используется в действии «Получить уникальные значения столбца», необходим ввод названий или номеров столбцов (с 1-цы), которые должны быть уникальными;
21. Для колонок – аналогично полю «Столбец»;
22. При каждом изменении в колонке – аналогично полю «Столбец»;
23. Лист, диапазон - поле, в которое должен быть введен лист файла (аналогично полю «Лист»), затем поставлена запятая и указан диапазон или ячейка (аналогично полю «Ячейки»);

24. Лист, диапазон источника – аналогично полю «Лист, диапазон»;
25. Лист, диапазон для вставки – аналогично полю «Лист, диапазон»;
26. Поля в столбцах – используется в действии «Сводная таблица», необходимо ввести название колонки из рабочей таблицы (не А, В, С и т.д., а именно название столбца вашей таблицы, например «Дата создания», без использования кавычек);
27. Поля в значениях – аналогично «Поля в столбцах»;
28. Поля в строках – аналогично «Поля в столбцах»;
29. Копируемый лист – аналогично полю «Лист»;
30. Копируемый диапазон/ячейка – аналогично полю «Ячейки»;
31. Диапазон/ячейка для вставки – аналогично полю «Ячейки»;
32. Файл для вставки (необяз.) – путь до файла с его расширением, является необязательным полем;
33. Расположение (лев. верхн., прав. нижн.) – две ячейки, разделенные запятой, которые обозначают положение - первая – левый верхний угол, вторая – правый нижний;
34. Диапазон названий – аналогично полю «Ячейки»;
35. Диапазон аргументов - аналогично полю «Ячейки»;
36. Диапазон значений - аналогично полю «Ячейки»;
37. Ячейка для вставки - аналогично полю «Ячейки», но только в единственном числе;
38. Тип данных – выпадающий список, обязательен выбор одного значения из представленных;
39. Операция – аналогично «Типу данных»;
40. Критерии (ч/з ;) – аргументы для выбранной операции, могут быть как числами, так и строками, перечисляются через «;»;
41. Цвет заливки – цвет в формате ARGB, с перечислением каждого параметра через запятую или в формате HEX (например, #FFFFFF), возможен выбор цвета через диалог выбора цвета по кнопке в поле:



Рис. 2.4.4. Поле «Цвет заливки»

42. Цвет текста – аналогично «Цвет заливки»;
43. Цвет – аналогично полю «Цвет заливки»;
44. Значение для сравнения – текстовое или числовое значение;

45. Ячейка/диапазон – аналогично полю «Ячейки»;
46. Столбцы – перечисление номеров (с 1-цы) или названий необходимых столбцов через запятую;
47. Ширина – число, как целое, так и дробное (через «.» или «,»);
48. Строки – перечисление номеров (с 1-цы) строк через запятую;
49. Высота – аналогично полю «Ширина»;
50. Общее положение – выпадающий список, обязателен выбор одного из предложенных значений;
51. Выравнивание содержимого – аналогично полю «Общее положение»;
52. Шрифт – выпадающий список со всеми шрифтами, имеющимися в Microsoft Office, необходимо выбрать один из них;
53. Стиль текста – аналогично полю «Общее положение»;
54. Размер – аналогично полю «Ширина»;
55. Расположение – аналогично полю «Общее положение»;
56. Стиль – аналогично полю «Общее положение».

Во всех полях можно использовать переменные, но в них должно содержаться то, что требуют поля, то есть, в поле «Столбец» можно написать v.column, но тогда в v.column должен содержаться номер столбца, начинающийся с единицы, либо его название. Поля, которые закрашены серым цветом, не используются в выбранном действии.

Глава 5. Описание действий, доступных в модуле

Действия разделены на группы для удобной навигации по выпадающему списку. Если вы знаете точное название действия, то вы можете начать вводить его в поле «Действие», и программа постараётся подсказать вам то действие, которое вы ищите.

Группа «Работа с данными»

Действие «Считать данные»

Данное действие позволяет считать значение из одной конкретной ячейки, либо из диапазона ячеек. Если будет указана одна ячейка, то создаётся переменная с типом «Элемент», если же диапазон, то будет создана переменная с типом «Таблица».

Добавить команду

Действие	Считать данные	Формула
Лист	Лист1	Функция
Ячейки	A1:B3	Переменная
		v.table

Рис. 2.5.1. Пример настройки действия «Считать данные»

Действие «Записать данные»

Действие позволяет записать указанное значение в одну или в диапазон ячеек.

Добавить команду

Действие	Записать данные	Формула
Лист	Лист1	Функция
Ячейки	A1	Значение
		Номер строки

Рис. 2.5.2. Пример настройки действия «Записать данные»

Действие «Формула»

Действие позволяет записать формулу в указанную ячейку (или в диапазон) и вернуть её результат в указанную переменную. Поле «Переменная» можно оставить пустым, если результат выполнения данной формулы не важен.

Добавить команду

Действие	Формула	Формула	=СУММ(А1:В1)
Лист	Лист1	Функция	
Ячейки	C1	Переменная	v.sum

Рис. 2.5.3. Пример настройки действия «Формула»

СОВЕТ. Предположим, что вам необходимо написать формулу, применимую к текущей строке, и «растянуть» её на несколько других строк, как в Excel. Начните с создания действия «Формула» для первой строки. Затем «растяните» её с помощью действия «Скопировать и вставить ячейку/диапазон» применительно к настроенной формуле. То есть для выше предложенной формулы «растягивание» будет выглядеть так:

Добавить команду

Действие	Скопировать и вставить ячейку/диапазон	Лист для вставки	Лист1
Копируемый лист	Лист1	Файл для вставки (необяз.)	
Копируемый диапазон/ячейка	C1	Диапазон/ячейка для вставки	C2:C10

Рис. 2.5.4. Пример настройки действия «Скопировать и вставить ячейку/диапазон»

Действие «Получить формулу»

Действие возвращает формулу из ячейки в указанную переменную.

Добавить команду

Действие	Получить формулу	Формула	
Лист	Лист1	Функция	
Ячейки	C1	Переменная	v.formula

Рис. 2.5.5. Пример настройки действия «Получить формулу»

Действие «Заменить формулу на её значение»

Данное действие заменяет формулу в ячейке или диапазоне ячеек на её результат.

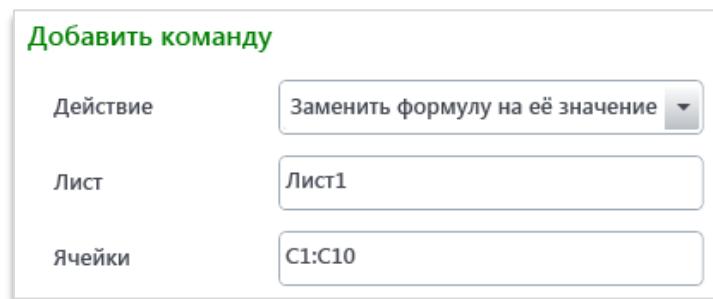


Рис. 2.5.6. Пример настройки действия «Заменить формулу на ее значение»

Группа «Работа со строками»

Действие «Добавить строку»

Действие вставляет новую строку в указанное место.

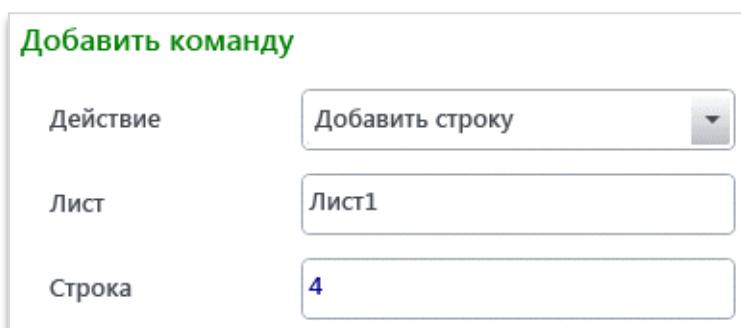


Рис. 2.5.7. Пример настройки действия «Добавление строки»

Действие «Удалить строку»

Действие удаляет указанную строку.

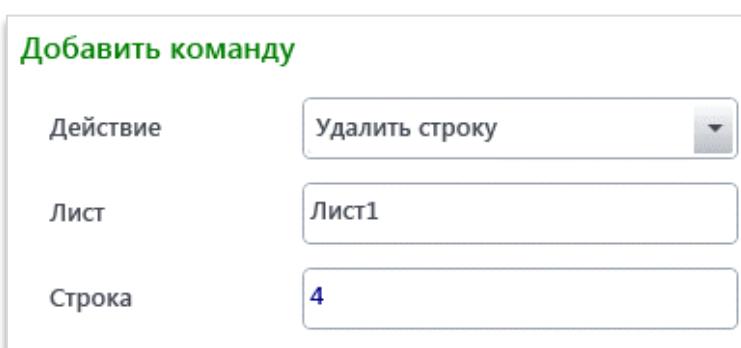


Рис. 2.5.8. Пример настройки действия «Удаление строки»

Действие «Получить количество строк»

Действие возвращает номер последней используемой строки в листе в указанную переменную.

Добавить команду

Действие	Получить количество строк	Формула
Лист	Лист1	Функция
Ячейки		Переменная
		v.countRow

Рис. 2.5.9. Пример настройки действия «Получение количества строк»

Действие «Получить количество отфильтрованных строк»

Действие возвращает количество видимых строк при примененном фильтре в указанную переменную.

Добавить команду

Действие	Получить количество отфильтрован	Формула
Лист	Лист1	Функция
Ячейки		Переменная
		v.countFilteredRow

Рис. 2.5.10. Пример настройки действия «Получение количества отфильтрованных строк»

Действие «Переместить строку»

Действие помещает указанную в поле «Перемещаемая строка» строку на листе из поля «Лист» вместо указанной в поле «Номер строки для вставки» листа «Лист для вставки».

Добавить команду

Действие	Переместить строку	Формула
Лист	Лист1	Лист для вставки
Перемещаемая строка	14	Номер строки для вставки
		16

Рис. 2.5.11. Пример настройки действия «Переместить строку»

В данном примере строка 14 с листа Лист1 будет перемещена на место 16 строки листа Лист2.

Действие «Сгруппировать строки»

С помощью данного действия создаются группы строк в Excel. На вход подается название листа, на котором происходит группировка, и перечень номеров строк. Перечислить строки можно просто через запятую, либо, если диапазон не разрывной, через двоеточие, например – «1,2,3,7,8,9» или «1:5», что равносильно «1,2,3,4,5».

Добавить команду

Действие	Сгруппировать строки
Лист	Лист1
Строки	11:18

Рис. 2.5.12. Пример настройки действия «Сгруппировать строки»

Действие «Удалить группировку строк»

Действие предназначено для удаления 1-го уровня существующей группировки по строкам. Входные данные аналогичны действию «Сгруппировать строки».

Добавить команду

Действие	Удалить группировку строк
Лист	Лист1
Строки	13,14,15

Рис. 2.5.13. Пример настройки действия «Удалить группировку строк»

Действие «Уровень группировки строки»

Действие позволяет получить уровень группировки выбранной строки и записать результат в указанную переменную.

Добавить команду

Действие	Уровень группировки строки	Формула
Лист	Лист1	Функция
Строка	13	Переменная
		v.levelGroup

Рис. 2.5.14. Пример настройки действия «Уровень группировки строки»

Группа «Работа с колонками»

Действие «Добавить столбец»

Действие добавляет новый столбец в указанное место.

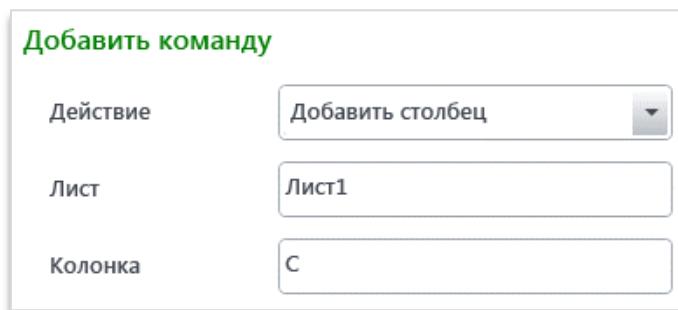


Рис. 2.5.15. Пример настройки действия «Добавление столбца»

Действие «Удалить столбец»

Действие удаляет указанный столбец.

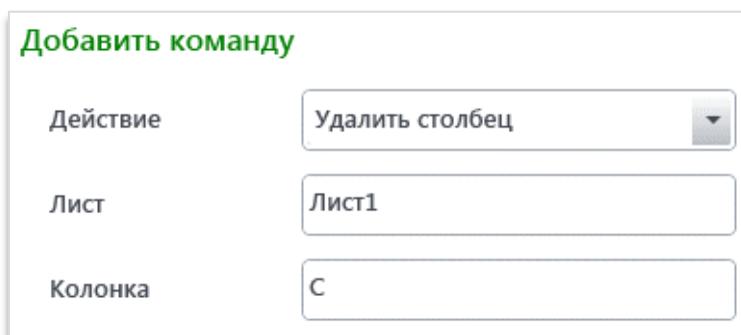


Рис. 2.5.16. Пример настройки действия «Удаление столбца»

Действие «Получить количество столбцов»

Действие возвращает номер самого последнего используемого столбца.

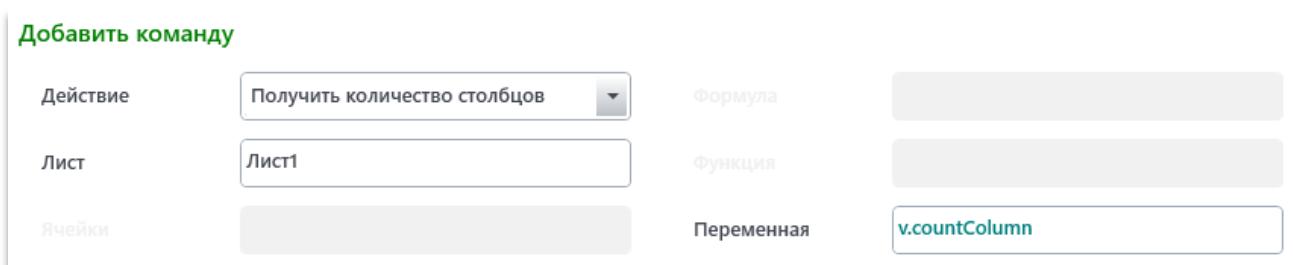


Рис. 2.5.17. Пример настройки действия «Получение количества столбцов»

Действие «Сгруппировать столбцы»

Действие объединяет указанные столбцы в группу. Входными данными являются лист и перечень столбцов через запятую или в виде диапазона через двоеточие («A,B,C,D,E» или

«A:E»). В качестве столбца можно указывать как его название, так и его порядковый номер, начиная с единицы.

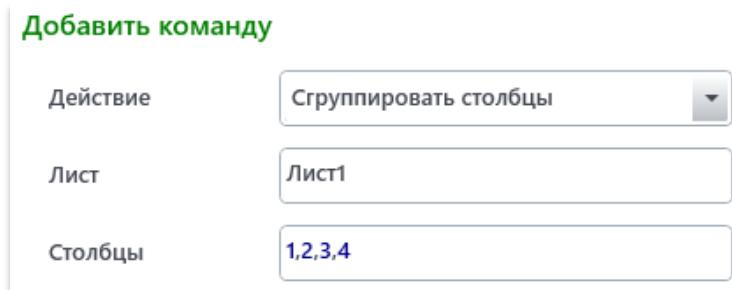


Рис. 2.5.18. Пример настройки действия «Сгруппировать столбцы»

Действие «Удалить группировку колонок»

Действие позволяет удалить один уровень группировки у указанных столбцов.

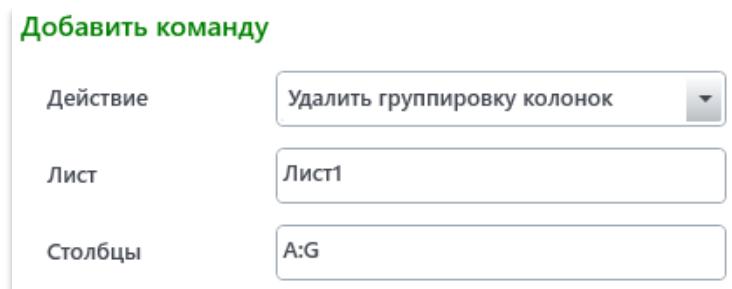


Рис. 2.5.19. Пример настройки действия «Удалить группировку колонок»

Действие «Уровень группировки столбца»

Данное действие позволяет получить уровень группы указанного столбца и записать его в переменную.

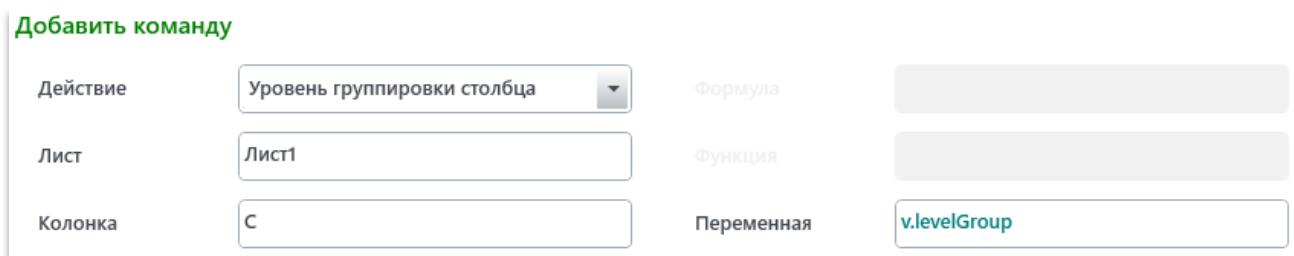


Рис. 2.5.20. Пример настройки действия «Уровень группировки столбца»

Группа «Работа с листами и файлами»

Действие «Создать новый лист»

Действие создает новый лист с указанным именем.

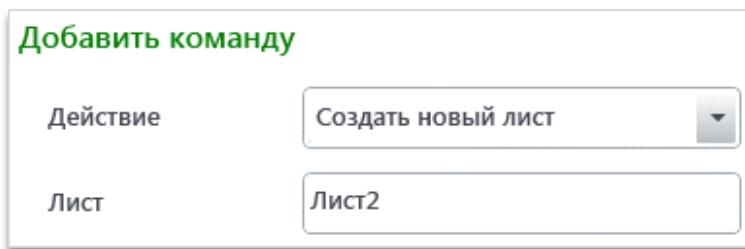


Рис. 2.5.21. Пример настройки действия «Создание нового листа»

Действие «Удалить лист»

Действие удаляет лист с указанным именем.

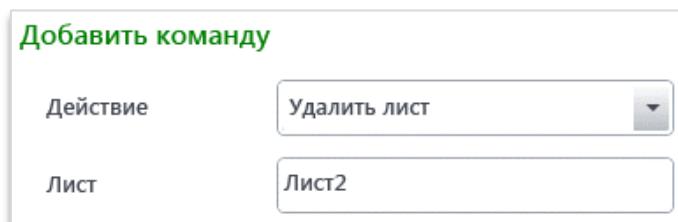


Рис. 2.5.22. Пример настройки действия «Удаление листа»

Действие «Очистить лист»

Действие очищает ячейки листа от данных. **Не очищает форматирование!**

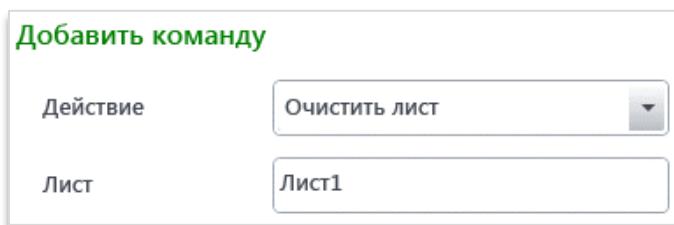


Рис. 2.5.23. Пример настройки действия «Очищение листа»

Действие «Получить список листов»

Возвращает переменную-список, содержащую названия всех листов файла.

Добавить команду

Действие	Получить список листов	Формула
Лист		Функция
Ячейки		Переменная
		v.listSheets

Рис. 2.5.24. Пример настройки действия «Получение списка листов»

Действие «Создать новый файл»

Действие создает в указанной папке новый файл с названием и указанным расширением (если расширение не указано, создается файл .xlsx). Если поле «Лист» заполнено, то первый лист файла будет назван так, как указано в поле, иначе - по умолчанию (Лист 1, Sheet 1, в зависимости от языка системы).

Добавить команду			
Действие	Создать новый файл	Формула	
Лист	Лист1	Функция	
Путь к файлу	C:\Users\Desktop\Робот	Имя файла	NewFile

Рис. 2.5.25. Пример настройки действия «Создание нового файла»

Действие «Заполнить лист на основе Excel-файла»

Данное действие производит заполнение листа текущего файла по аналогии с листом какого-либо другого файла. В поле «Лист» указывается лист текущего файла.

Добавить команду			
Действие	Заполнить лист на основе Excel-файла	Лист копируемого файла	Лист1
Лист	Лист2	Функция	
Путь копируемого файла	C:\Users\Desktop\Робот\File.xlsx	Значение	

Рис. 2.5.26. Пример настройки действия «Заполнение листа»

Действие «Экспортировать в PDF»

Действие позволяет сохранить Excel-файл в формате PDF стандартными методами Windows, то есть деление листа Excel-файла на листы в документе PDF будет производиться по стандартным настройкам, указанным в файле Excel (по умолчанию размер листа A4, с обычными полями). Действие не имеет дополнительной настройки.

Добавить команду

Действие	Экспортировать в PDF	Формула
Лист		Функция
Ячейки		Значение

Рис. 2.5.27. Пример настройки действия «Экспортирование в PDF»

Название файла и его расположение будут совпадать с экспортируемым файлом (расширение файла будет pdf).

Группа «Поиск»

Действие «Найти данные»

Ищет заданную строку в указанном листе и возвращает адрес *первой* найденной ячейки. Флажок «Строгое сравнение» ставится в том случае, если текст в ячейке должен полностью совпадать с текстом, введенном в поле «Строка».

Добавить команду

Действие	Найти данные	Формула
Лист	Лист1	Строгое сравнение <input checked="" type="checkbox"/>
Строка	Дата	Переменная <code>v.findData</code>

Рис. 2.5.28. Пример настройки действия «Найти данные»

Действие «Поиск строки по значениям колонок»

Данное действие ищет и возвращает номера строк, в заданных колонках которых записаны указанные значения. Флажок «Строгое сравнение» ставится в том случае, если текст в ячейках должен полностью совпадать с текстом, введенном в поле «Значения». Флажок «Массив» вернет массив всех найденных строк, результирующая переменная будет являться массивом чисел, если флажок не установлен, то в результирующую переменную будет записан номер первой найденной строки или «Строка не найдена», если таких строк нет.

Добавить команду

Действие	Поиск строки по значениям колонок	Значения (ч/з ;)	23.03.2020; 'Уфа'
Лист	Лист1	Строгое сравнение	<input checked="" type="checkbox"/>
Колонки (ч/з ;)	A;B	Массив	<input type="checkbox"/>
		Переменная	<code>v.findData</code>

Рис. 2.5.29. Пример настройки действия «Поиск строки»

В данном случае будет искааться одна строка, в колонке А которой 23.03.2020, а в колонке В – Уфа.

Действие «Найти лист»

Ищет лист, в котором есть ячейка с указанным в поле «Строка» значением и возвращает его название в указанную переменную. Флажок «Строгое сравнение» ставится в том случае, если текст в ячейке должен полностью совпадать с текстом, введенном в поле «Строка».

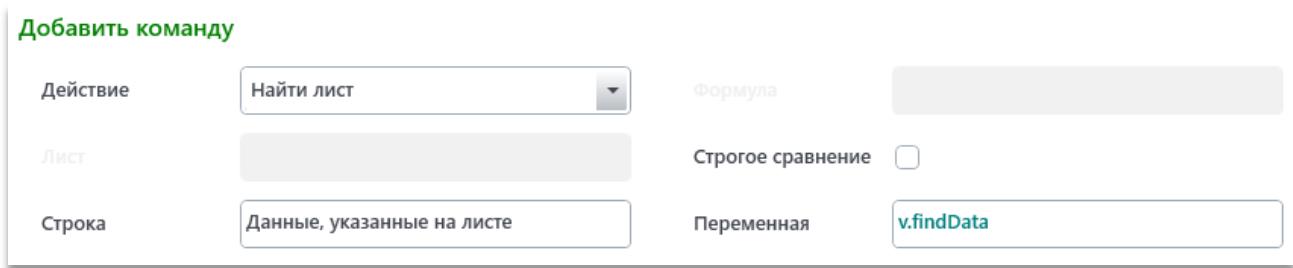


Рис. 2.5.30. Пример настройки действия «Найти лист»

Группа «Другие функции»

Действие «Сместить ячейку»

Смещает заданную ячейку на указанное количество строк и колонок, затем возвращает результат в переменную (в данном примере результатом будет ячейка D13).

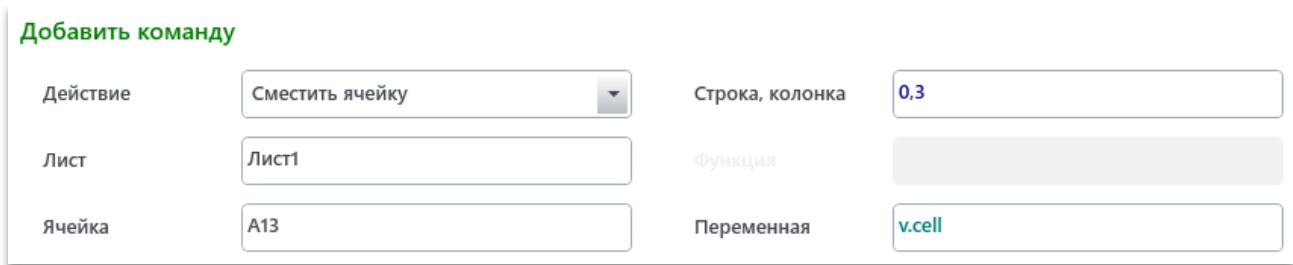


Рис. 2.5.31. Пример настройки действия «Сместить ячейку»

Действие «Наложить фильтр»

Данное действие добавляет фильтр в таблицу на указанном листе, ячейка из которой указана в поле «Ячейка из диапазона значений». В поле столбец записывается столбец, к которому необходимо применить фильтр. Фильтруемый столбец должен быть один. Если необходимо применить несколько фильтров к одной таблице, то необходимо добавить для каждого фильтра свое действие в рамках одного Excel-модуля. Функция, по которой должно происходить сравнение, выбирается из выпадающего списка в поле «Функция», значение, с которым будет происходить сравнение при фильтрации, указано в поле «Значение».

Добавить команду

Действие	Наложить фильтр	Значение	15
Лист	Лист 2	Функция	больше
Колонка	A	Ячейка из диапазона значений	A2

Рис. 2.5.32. Пример настройки действия «Наложить фильтр»

Действие «Удалить фильтр»

Действие удаляет все фильтры с листа.

Добавить команду

Действие	Удалить фильтр
Лист	Лист 2
Ячейки	

Рис. 2.5.33. Пример настройки действия «Удаление фильтра»

Действие «Добавить сортировку»

Действие добавляет сортировку к указанному диапазону по выбранному столбцу. В зависимости от флагка определяется, сортируется столбец по убыванию или по возрастанию. Столбец для сортировки должен быть один.

Добавить команду

Действие	Добавить сортировку	Колонка	B
Лист	Лист 2	Функция	
Ячейки	A1:D15	По убыванию	<input type="checkbox"/>

Рис. 2.5.34. Пример настройки действия «Добавление сортировки»

Действие «Очистить диапазон»

Действие удаляет все значения и форматирования ячеек в указанном диапазоне.

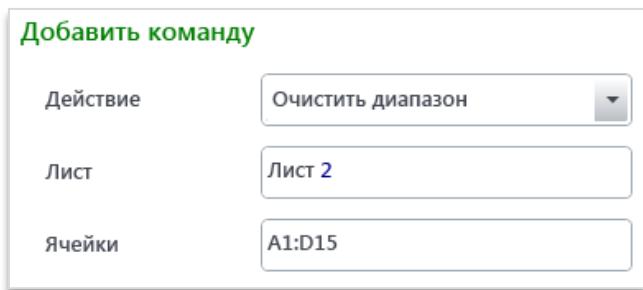


Рис. 2.5.35. Пример настройки действия «Очищение диапазона»

Действие «Получить уникальные значения столбца»

Аналогично действию «Удалить дубликаты» в Excel, данное действие возвращает только уникальные значения указанных столбцов из диапазона. Итоговая таблица может быть вставлена как в тот же лист, так и в другой в лист, но в тот же диапазон. В зависимости от наличия дубликатов, количество строк может быть уменьшено.

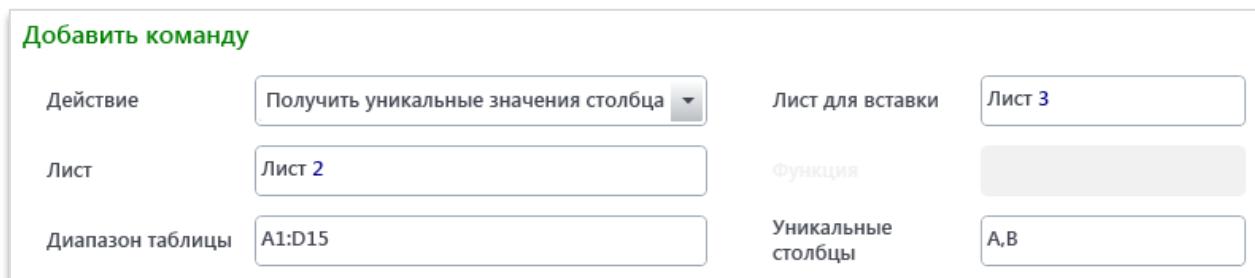


Рис. 2.5.36. Пример настройки действия «Получение уникальных значений»

Действие «Промежуточные итоги»

Действие добавляет операцию Excel «Промежуточные итоги» на указанный лист к указанному диапазону.

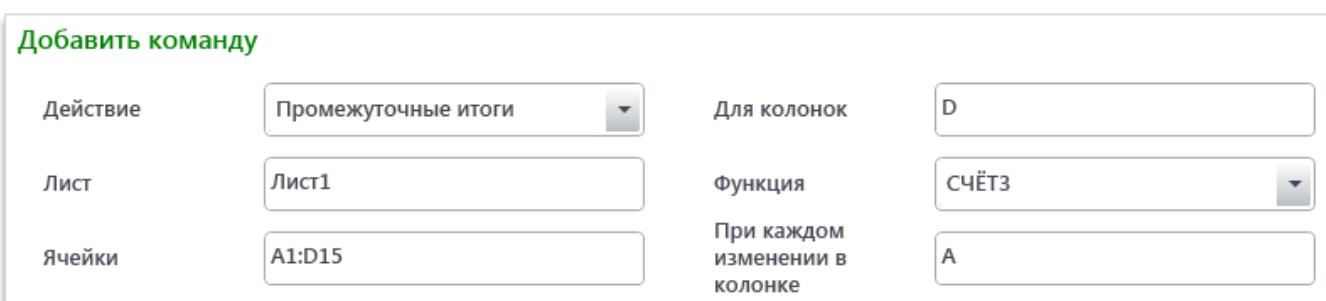


Рис. 2.5.37. Пример настройки действия «Промежуточные итоги»

Поле «Для колонок» соответствует полю в Excel «Добавить итоги по» (выделено зеленым на Рис. 2.5.38), поле «При каждом изменении в колонке» соответствует аналогичному полю в Excel, но должно быть заполнено названием колонки Excel (A, B, ... или 1, 2, ...) (выделено

красным на Рис. 2.5.38, поле «Функция» заполняется операцией, которая должна быть применена для выбранных колонок (выделено синим на Рис. 2.5.38).

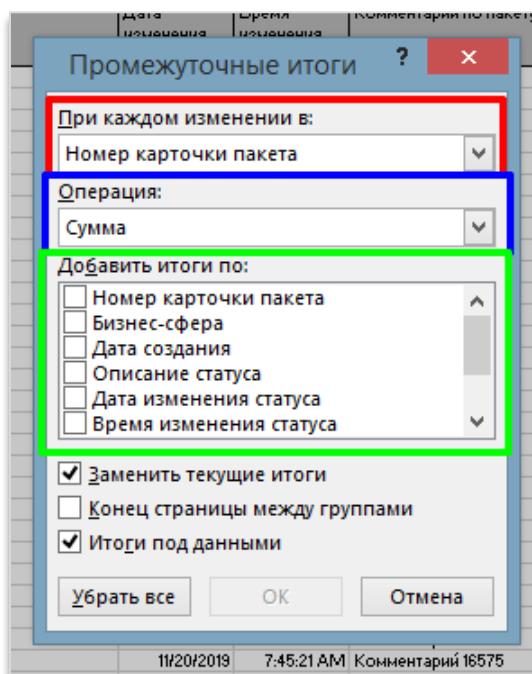


Рис. 2.5.38. Настройка действия «Промежуточные итоги» в Excel

Действие «Сводная таблица»

Добавляет сводную таблицу в указанные лист и диапазон (поле «Лист, диапазон для вставки») на основе данных таблицы из поля «Лист, диапазон источника».

Добавить команду			
Действие	Сводная таблица	Поля в столбцах	Бизнес-сфера
Лист, диапазон источника	Лист1, A1:E200	Поля в строках	Дата создания;Номер карточки пакета
Лист, диапазон для вставки	Лист2, A1	Поля в значениях	Нарушение срока

Рис. 2.5.39. Пример настройки действия «Сводная таблица»

В поля «Поля в столбцах», «Поля в значениях» и «Поля в строках» вписываются **названия колонок таблицы-источника** (а не Excel-таблицы), как в программе Excel:

«Поля в столбцах» соответствуют названиям столбцов таблицы, значения которых должны быть расположены в столбцах сводной таблицы (выделено зеленым на Рис. 2.5.40);

«Поля в строках» соответствуют названиям столбцов таблицы, значения которых должны быть расположены в строках сводной таблицы (выделено синим на Рис. 2.5.40);

«Поля в значениях» соответствуют названиям столбцов таблицы, значения которых должны быть расположены в значениях сводной таблицы (выделено красным на Рис. 2.5.40);

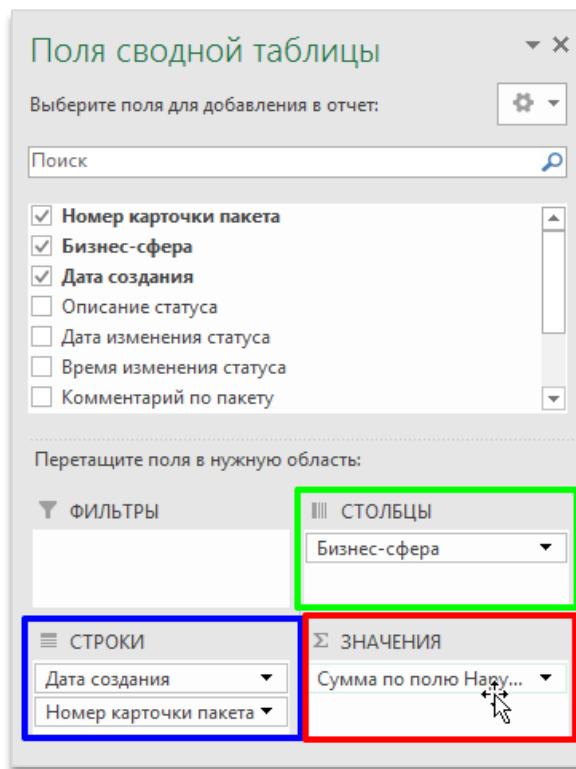


Рис. 2.5.40. Настройка действия «Сводная таблица» в Excel

Действие «Скопировать и вставить ячейку/диапазон»

Данное действие позволяет скопировать данные и форматирование и вставить их в тот же или в другой файл. С помощью него можно «растягивать» формулы (см. «Действие «Формула»).

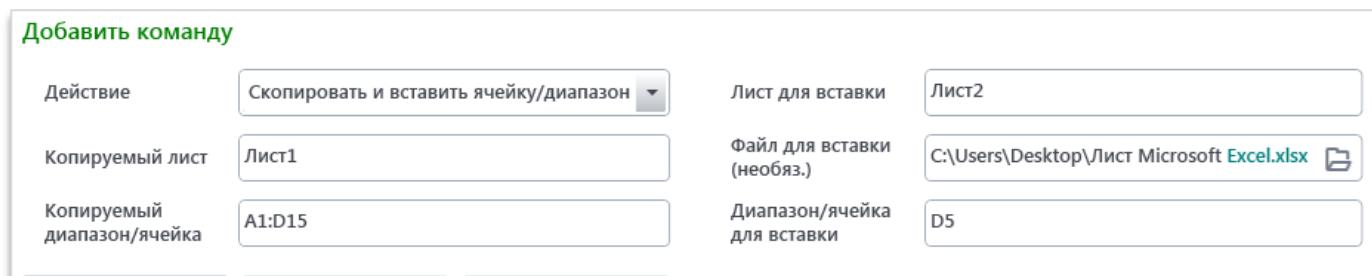


Рис. 2.5.41. Пример настройки действия «Скопировать и вставить ячейку/диапазон»

Действие «Отобразить в виде процентов»

Действие переводит выбранный диапазон в проценты.

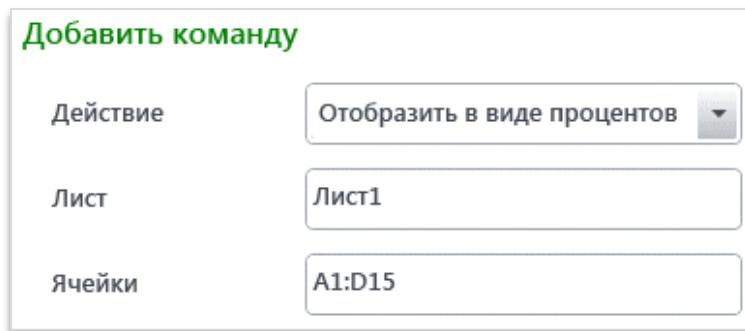


Рис. 2.5.42. Пример настройки действия «Отображение в виде процентов»

Действие «Круговая диаграмма»

Действие создает на указанном листе круговую диаграмму. Значения диаграммы берутся из диапазона, указанного в поле «Ячейки», диапазон с названиями значений указывается в поле «Диапазон названий». В поле «Расположение» указываются границы расположения круговой диаграммы, а именно левый верхний угол и правый нижний через запятую.

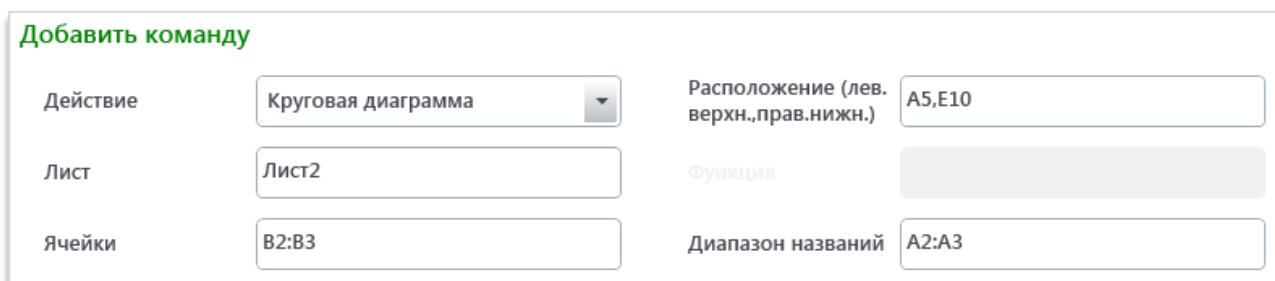


Рис. 2.5.43. Пример настройки действия «Создание круговой диаграммы»

ПРИМЕР.

Построение круговой диаграммы для двух параметров, указанных в столбце А, по соответствующим им параметрам, указанным в столбце В. Заполнение действия для данного примера указано на Рис. 2.5.43.

Пример исходных данных представлен на Рис. 2.5.44.

	A	B	C
1			
2	Первый параметр	1002	
3	Второй параметр	580	
4			
5			
6			
7			
8			

Рис. 2.5.44. Пример исходных данных

Результат построения круговой диаграммы представлен на Рис. 2.5.45.

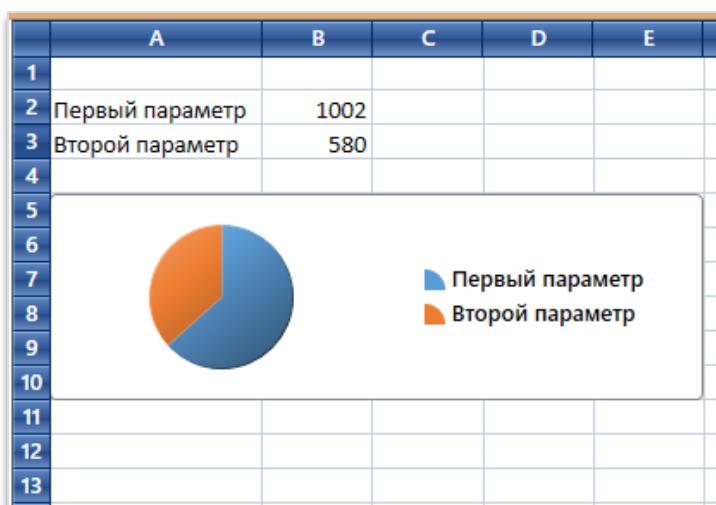


Рис. 2.5.45. Результат выполнения действия «Круговая диаграмма»

Действие «График»

Действие создает линейный график с одним или несколькими рядами. Ряд необходимо указать в поле «Диапазон значений» в виде одномерного диапазона (по одной колонке или по одной строке). Если рядов больше, чем один, то последующие ряды нужно указать в таком же виде через «;», как указано на Рис. 2.5.46. Диапазон аргументов должен быть один и указывается аналогично ряду. «Ячейки с названиями рядов» заполняются названиями ячеек через «;», их количество должно быть равно количеству рядов, первая ячейка отображает название первого ряда и т.д. В ячейку для вставки записывается ячейка, в которой будет располагаться левый верхний угол графика.

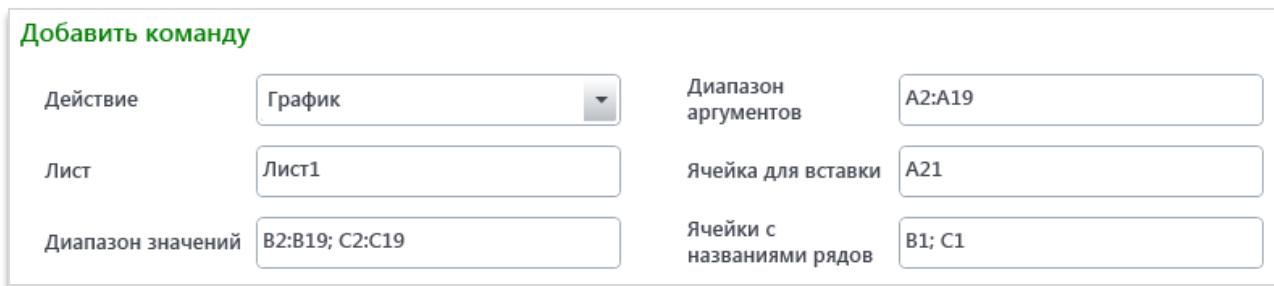


Рис. 2.5.46. Пример настройки действия «График»

ПРИМЕР.

В качестве примера возьмем курс доллара и евро за некоторый промежуток времени. Так как валют две, то график должен состоять из двух рядов. В качестве аргументов выступит столбец с указанием дат – А, в качестве значений будет два столбца, первый – В, курс евро, и второй – С, курс доллара. Заполнение действия представлено на Рис. 2.5.46.

Пример исходных данных представлен на Рис. 2.5.47.

	A	В	С
1	Дата	Курс евро	Курс доллара
2	24.03.2020	86,705	78,0443
3	21.03.2020	84,1552	80,157
4	20.03.2020	87,2669	77,2131
5	19.03.2020	84,8881	73,8896
6	18.03.2020	82,3056	74,1262
7	17.03.2020	82,7471	73,1882
8	14.03.2020	81,861	74,0274
9	13.03.2020	83,6584	71,472
10	12.03.2020	81,0207	72,0208
11	11.03.2020	81,8588	67,5175
12	07.03.2020	75,8424	66,1854
13	06.03.2020	73,6842	66,0784
14	05.03.2020	73,7369	66,4437
15	04.03.2020	73,9385	66,3274
16	03.03.2020	73,4178	66,9909
17	29.02.2020	73,7235	65,6097
18	28.02.2020	71,6458	65,5177
19	27.02.2020	71,2439	64,9213
20			

Рис. 2.5.47. Пример настройки действия «График»

Результат работы программы представлен на Рис. 2.5.48.

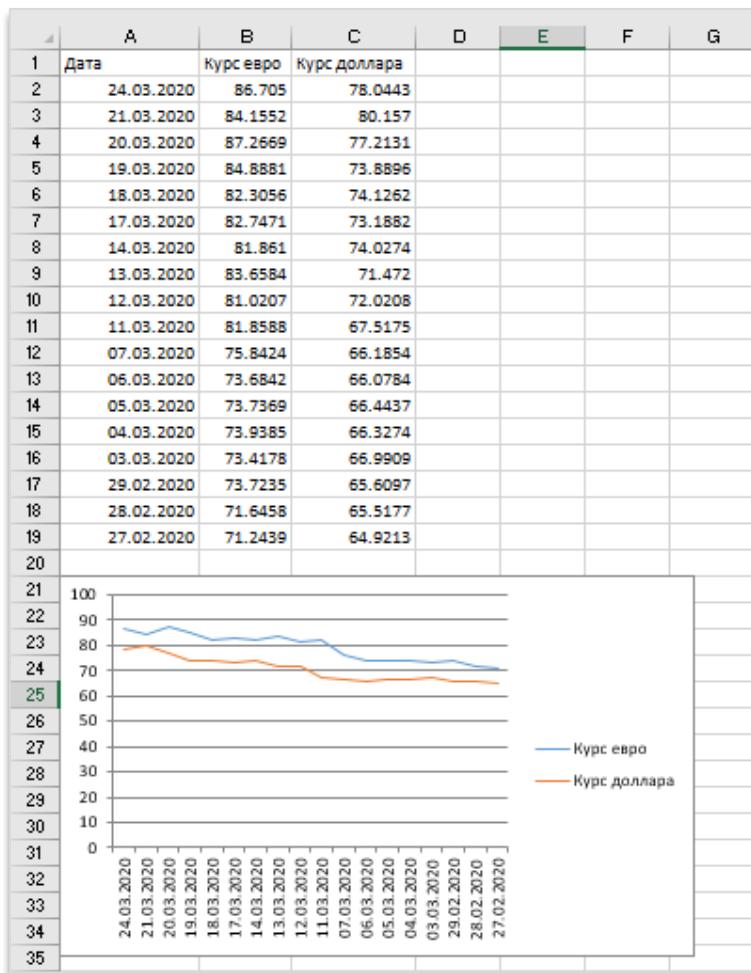


Рис. 2.5.48. Результат выполнения действия «График»

Действие «Проверка значений»

Действие создает ограничение типов данных, которые можно ввести в выбранные ячейки, путем выбора предложенных правил. В поле «Лист» вводится лист, в ячейки которого добавляются ограничения, в поле «Ячейки» вводится диапазон ячеек, на которых будет проводиться проверка значений при вводе. В поле «Тип данных» выбирается один из предложенных типов данных – целое или действительное число, список, дата, длина текста, время. Поле «Операция» состоит из списка доступных операций для составления правила проверки – равно, не равно, больше, меньше, больше или равно, меньше или равно, между и вне. При использовании типа данных «Список» заполнение поля «Операция» не требуется, поэтому оно не доступно для редактирования.

Последнее поле меняется в зависимости от типа данных – для типа «Список» это будет «Источник данных», для всех остальных – «Критерии (ч/з ;)». Поле «Источник данных» заполняется диапазоном значений (внутри одного столбца или одной строки), в которой будет указан в списке (если список статичный, то указывайте абсолютную ссылку). Поле «Критерии (ч/з ;)» заполняется значениями, относительно которых будет применяться правило проверки.

Поле «Сообщение об ошибке» заполняется текстом ошибки, которая будет отображаться, если в ячейку введено значение, не удовлетворяющее заданному для неё правилу. Является необязательным полем. Если оставить его пустым, то будет отображаться ошибка «Введенное значение неверно. Набор значений, которые могут быть введены в ячейку, ограничен».

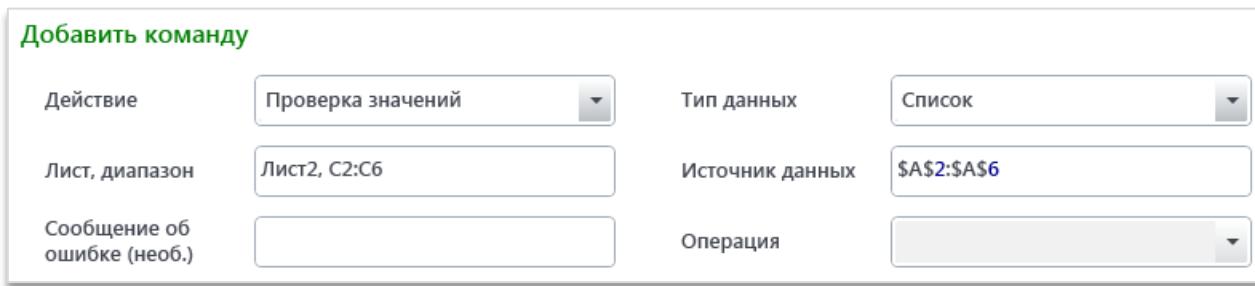


Рис. 2.5.49. Пример настройки действия «Проверка значений»

ПРИМЕР.

Для создания списка, состоящего из некоторых заданных в какой-либо колонке значений, действие заполняется так, как указано на Рис. 2.5.49. Пусть в ячейках A2:A6 будут следующие значения:

	A	B	C
1			
2	Первый параметр		
3	Второй параметр		
4	Третий параметр		
5	Четвертый параметр		
6	Пятый параметр		
7			

Рис. 2.5.50. Пример исходных данных

Тогда после работы робота при нажатии на любую ячейку из диапазона C2:C6 будет появляться стрелочка для открытия выпадающего списка, в котором будут содержаться выбранные значения:

	A	B	C	D
1				
2	Первый параметр			
3	Второй параметр			
4	Третий параметр			
5	Четвертый параметр			
6	Пятый параметр			
7				
8				
9				
10				
11				
12				

Рис. 2.5.51. Результат выполнения действия «Проверка значений»

Второй пример - создание правила, позволяющего вводить в диапазон ячеек A1:A5 только действительные числа, находящиеся в промежутке между 5 и 10. Пусть при введении неверного значения будет выдаваться ошибка: «Введено неверное значение. Должно быть введено действительное число между 5 и 10»:

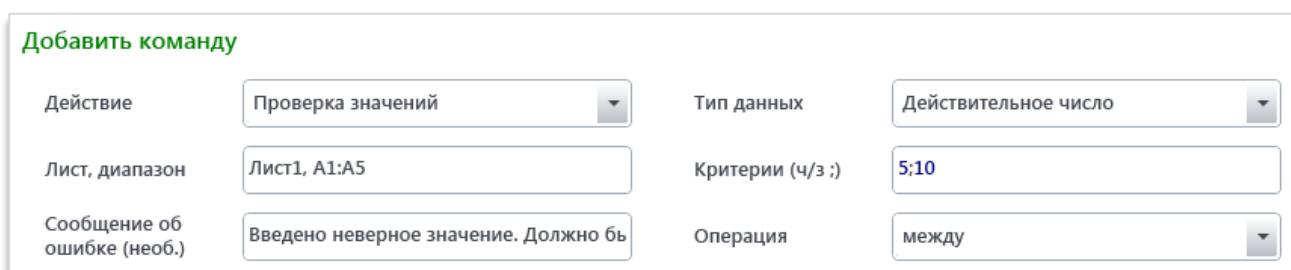


Рис. 2.5.52. Пример создания правила для проверки значений

После работы робота при попытке ввести число, находящееся вне диапазона от 5 до 10, появляется ошибка:

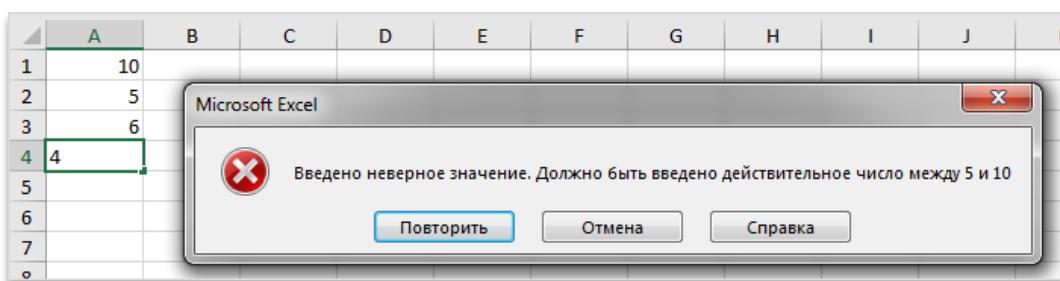


Рис. 2.5.53. Результат выполнения действия «Проверка значений»

Действие «Добавить изображение»

Действие добавляет выбранное изображение в текущий Excel-файл. Если изображение имеет прозрачный фон, то оно также сохранит прозрачность при его добавлении в файл Excel. В поле «Ячейки» указывается левый верхний угол расположения изображения.

Добавить команду

Действие	Добавить изображение	Формула	
Лист	Лист1	Функция	
Ячейки	B2	Путь до изображения	C:\Users\Desktop\Безымянный.png 

Рис. 2.5.54. Пример настройки действия «Добавить изображение»

Группа «Форматирование»

Действие «Условное форматирование»

Действие форматирует указанные ячейки на выбранном листе в зависимости от того, удовлетворяют ли они указанному условию. В поле «Лист, диапазон» через запятую вводятся лист и диапазон или одна ячейка из этого листа, к которым будет применяться условное форматирование. В полях «Цвет заливки» и «Цвет текста» указываются цвета, которыми будет залита ячейка и окрашен текст в ней, если она будет удовлетворять условию. Условие составляется из выбора функции и указания значения для сравнения. Доступные функции – равно, не равно, больше, меньше, больше или равно, меньше или равно, формула. Если выбрана функция «формула», то в поле «Значение для сравнения» следует вписать формулу в формате Excel, начиная со знака «=».

Добавить команду

Действие	Условное форматирование	Цвет текста	#000000  
Лист, диапазон	Лист1, B2:C19	Функция	меньше или равно
Цвет заливки	#00FF00  	Значение для сравнения	82

Рис. 2.5.55. Пример настройки действия «Условное форматирование»

ПРИМЕР.

Окрасим все ячейки в столбцах В и С в зеленый цвет, если значение курса меньше или равно 82. Действие заполняется как на Рис. 2.5.55.

Исходные данные представлены на Рис. 2.5.56.

	A	B	C
1	Дата	Курс евро	Курс доллара
2	24.03.2020	86,705	78,0443
3	21.03.2020	84,1552	80,157
4	20.03.2020	87,2669	77,2131
5	19.03.2020	84,8881	73,8896
6	18.03.2020	82,3056	74,1262
7	17.03.2020	82,7471	73,1882
8	14.03.2020	81,861	74,0274
9	13.03.2020	83,6584	71,472
10	12.03.2020	81,0207	72,0208
11	11.03.2020	81,8588	67,5175
12	07.03.2020	75,8424	66,1854
13	06.03.2020	73,6842	66,0784
14	05.03.2020	73,7369	66,4437
15	04.03.2020	73,9385	66,3274
16	03.03.2020	73,4178	66,9909
17	29.02.2020	73,7235	65,6097
18	28.02.2020	71,6458	65,5177
19	27.02.2020	71,2439	64,9213
20			

Рис. 2.5.56. Пример исходных данных

Результат работы программы представлен на Рис. 2.5.57.

	A	B	C
1	Дата	Курс евро	Курс доллара
2	24.03.2020	86,705	78,0443
3	21.03.2020	84,1552	80,157
4	20.03.2020	87,2669	77,2131
5	19.03.2020	84,8881	73,8896
6	18.03.2020	82,3056	74,1262
7	17.03.2020	82,7471	73,1882
8	14.03.2020	81,861	74,0274
9	13.03.2020	83,6584	71,472
10	12.03.2020	81,0207	72,0208
11	11.03.2020	81,8588	67,5175
12	07.03.2020	75,8424	66,1854
13	06.03.2020	73,6842	66,0784
14	05.03.2020	73,7369	66,4437
15	04.03.2020	73,9385	66,3274
16	03.03.2020	73,4178	66,9909
17	29.02.2020	73,7235	65,6097
18	28.02.2020	71,6458	65,5177
19	27.02.2020	71,2439	64,9213
20			

Рис. 2.5.57. Результат выполнения действия «Условное форматирование»

Действие «Цвет ячейки/диапазона»

Действие меняет цвет заливки текста и самого текста в указанном диапазоне / ячейке.

Добавить команду

Действие	Цвет ячейки/диапазона	Цвет заливки	#00FFFF	 
Лист	Лист1	Функция		
Ячейка/диапазон	A19:B19	Цвет текста	#800080	 

Рис. 2.5.58. Пример настройки действия «Цвет ячейки/диапазона»

ПРИМЕР.

На Рис. 2.5.58 представлено, как залить диапазон ячеек A17:C19 в голубой цвет, а текст в этих ячейках окрасить в фиолетовый.

Исходные данные представлены на Рис. 2.5.59:

	A	В	С
1	Дата	Курс евро	Курс доллара
2	24.03.2020	86,705	78,0443
3	21.03.2020	84,1552	80,157
4	20.03.2020	87,2669	77,2131
5	19.03.2020	84,8881	73,8896
6	18.03.2020	82,3056	74,1262
7	17.03.2020	82,7471	73,1882
8	14.03.2020	81,861	74,0274
9	13.03.2020	83,6584	71,472
10	12.03.2020	81,0207	72,0208
11	11.03.2020	81,8588	67,5175
12	07.03.2020	75,8424	66,1854
13	06.03.2020	73,6842	66,0784
14	05.03.2020	73,7369	66,4437
15	04.03.2020	73,9385	66,3274
16	03.03.2020	73,4178	66,9909
17	29.02.2020	73,7235	65,6097
18	28.02.2020	71,6458	65,5177
19	27.02.2020	71,2439	64,9213
20			

Рис. 2.5.59. Пример исходных данных

Результат работы робота представлен на Рис. 2.5.60:

	A	В	С
1	Дата	Курс евро	Курс доллара
2	24.03.2020	86,705	78,0443
3	21.03.2020	84,1552	80,157
4	20.03.2020	87,2669	77,2131
5	19.03.2020	84,8881	73,8896
6	18.03.2020	82,3056	74,1262
7	17.03.2020	82,7471	73,1882
8	14.03.2020	81,861	74,0274
9	13.03.2020	83,6584	71,472
10	12.03.2020	81,0207	72,0208
11	11.03.2020	81,8588	67,5175
12	07.03.2020	75,8424	66,1854
13	06.03.2020	73,6842	66,0784
14	05.03.2020	73,7369	66,4437
15	04.03.2020	73,9385	66,3274
16	03.03.2020	73,4178	66,9909
17	29.02.2020	73,7235	65,6097
18	28.02.2020	71,6458	65,5177
19	27.02.2020	71,2439	64,9213
20			

Рис. 2.5.60. Результат выполнения действия «Цвет ячейки/диапазона»

Действие «Ширина столбца(-ов)»

Действие изменяет ширину указанных столбцов. Столбцы можно перечислять через запятую, писать их номера вместо названий (с 1-цы). Ширина указывается в той же единице измерения, которая используется в Excel.

Добавить команду

Действие	Ширина столбца(-ов)	Формула
Лист	Лист1	Функция
Столбцы	B,C	Ширина
		16

Рис. 2.5.61. Пример настройки действия «Ширина столбца(-ов)»

ПРИМЕР.

На Рис. 2.5.61 представлено, как заполнить действие для установки ширины столбцов В и С в 16 пунктов. В поле «Столбцы» можно было бы написать «2,3», то есть второй и третий столбцы.

Исходные данные представлены на Рис. 2.5.62:

	A	B	C
1	Дата	Курс евро	Курс доллара
2	24.03.2020	86,705	78,0443
3	21.03.2020	84,1552	80,157
4	20.03.2020	87,2669	77,2131

Рис. 2.5.62. Пример исходных данных

Результат работы представлен на Рис. 2.5.63:

	A	B	C
1	Дата	Курс евро	Курс доллара
2	24.03.2020	86,705	78,0443
3	21.03.2020	84,1552	80,157
4	20.03.2020	87,2669	77,2131

Рис. 2.5.63. Результат выполнения действия «Ширина столбца(-ов)»

Действие «Высота строки (-ок)»

Действие изменяет высоту указанных строк. Используется та же единица измерения, что и в Excel.

Добавить команду

Действие	Высота строки(-ок)	Формула
Лист	Лист1	Функция
Строки	1	Высота
		30

Рис. 2.5.64. Пример настройки действия «Высота строки(-ок)»

ПРИМЕР.

Изменим высоту первой строки, сделав её равной 30 пунктам. Для этого заполняем действие так, как указано на Рис. 2.5.64.

Исходные данные представлены на Рис. 2.5.65:

	A	B	C
1	Дата	Курс евро	Курс доллара
2	24.03.2020	86,705	78,0443
3	21.03.2020	84,1552	80,157
4	20.03.2020	87,2669	77,2131
5	19.03.2020	84,0001	77,0006

Рис. 2.5.65. Пример исходных данных

Результат работы робота представлен на Рис. 2.5.66:

	A	B	C
1	Дата	Курс евро	Курс доллара
2	24.03.2020	86,705	78,0443
3	21.03.2020	84,1552	80,157
4	20.03.2020	87,2669	77,2131

Рис. 2.5.66. Результат выполнения действия «Высота строки(-ок)»

Действие «Объединить ячейки»

Действие объединяет указанные ячейки на выбранном листе. В модуле предусмотрено три вида объединений:

- простое объединение – то есть объединяются и колонки, и столбцы диапазона;
- объединение по строкам;
- объединение по столбцам.

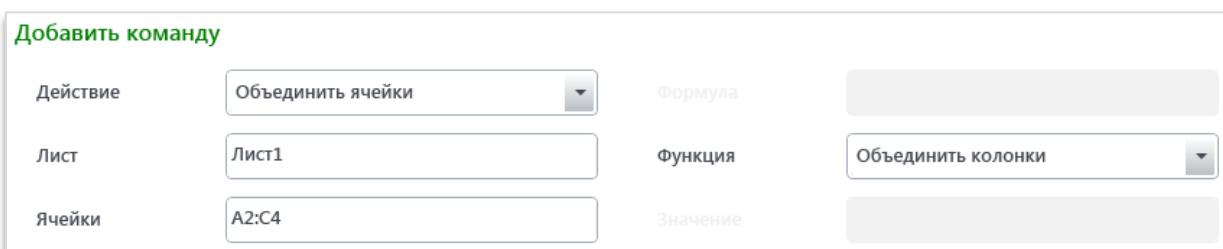


Рис. 2.5.67. Пример настройки действия «Объединить ячейки»

ПРИМЕР.

На Рис. 2.5.67 представлено заполнение действия для объединения колонок в диапазоне ячеек A2:C4.

Исходные данные представлены на Рис. 2.5.68:

	A	B	C
1	Дата	Курс евро	Курс доллара
2	24.03.2020	86,705	78,0443
3	21.03.2020	84,1552	80,157
4	20.03.2020	87,2669	77,2131
5	19.03.2020	84,8881	73,8896
6	18.03.2020	82,3056	74,1262
7	17.03.2020	82,7471	73,1882
8	14.03.2020	81,861	74,0274
9	13.03.2020	83,6584	71,472

Рис. 2.5.68. Пример исходных данных

Результат работы представлен на Рис. 2.5.69:

	A	B	C
1	Дата	Курс евро	Курс доллара
2			
3			
4	24.03.2020	86,705	78,0443
5	19.03.2020	84,8881	73,8896

Рис. 2.5.69. Результат выполнения действия «Объединить ячейки»

Действие «Очистить форматирование»

Действие очищает форматирование в указанном диапазоне выбранного листа.

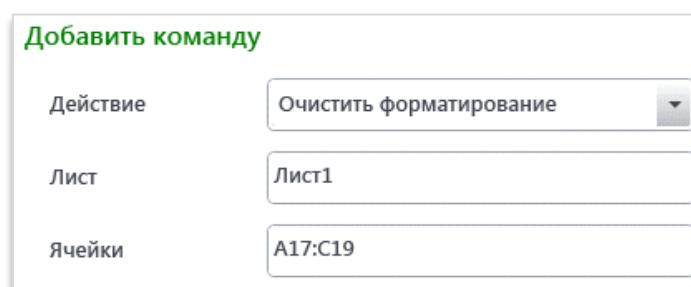


Рис. 2.5.70. Пример настройки действия «Очистить форматирование»

ПРИМЕР.

Очистим форматирование, которое содержится в диапазоне A17:C19. Для этого заполним действие так, как указано на Рис. 2.5.70 и запустим робота.

Исходные данные представлены на Рис. 2.5.71:

16	03.03.2020	73,4178	66,9909
17	29.02.2020	73,7235	65,6097
18	28.02.2020	71,6458	65,5177
19	27.02.2020	71,2439	64,9213
20			

Рис. 2.5.71. Пример исходных данных

Результат работы представлен на Рис. 2.5.72:

16	03.03.2020	73,4178	66,9909
17	29.02.2020	73,7235	65,6097
18	28.02.2020	71,6458	65,5177
19	27.02.2020	71,2439	64,9213
20			

Рис. 2.5.72. Результат выполнения действия «Очистить форматирование»

Действие «Формат ячейки»

Действие меняет формат указанной ячейки на выбранный. По умолчанию, все значения, заносимые в Excel студией, имеют строковый формат, поэтому, для корректного пересчета формул или сортировки, необходимо привести данные к нужному формату.

Добавить команду

Действие	Формат ячейки	Тип данных	Дата
Лист	Лист2	Функция	
Ячейки	A2:A6	Значение	

Рис. 2.5.73. Пример настройки действия «Формат ячейки»

Действие «Формат числа»

Действие отображает число по определенным правилам числового формата Excel.

Добавить команду

Действие	Формат числа	Формула	
Лист	Лист3	Функция	
Ячейки	A1	Значение	\$#,##0.00

Рис. 2.5.74. Пример настройки действия «Формат числа»

Примеры форматов можно посмотреть в программе Excel, кликнув правой кнопкой левой кнопкой мыши по ячейке – «Формат ячейки» – вкладка «Число» - (все форматы):

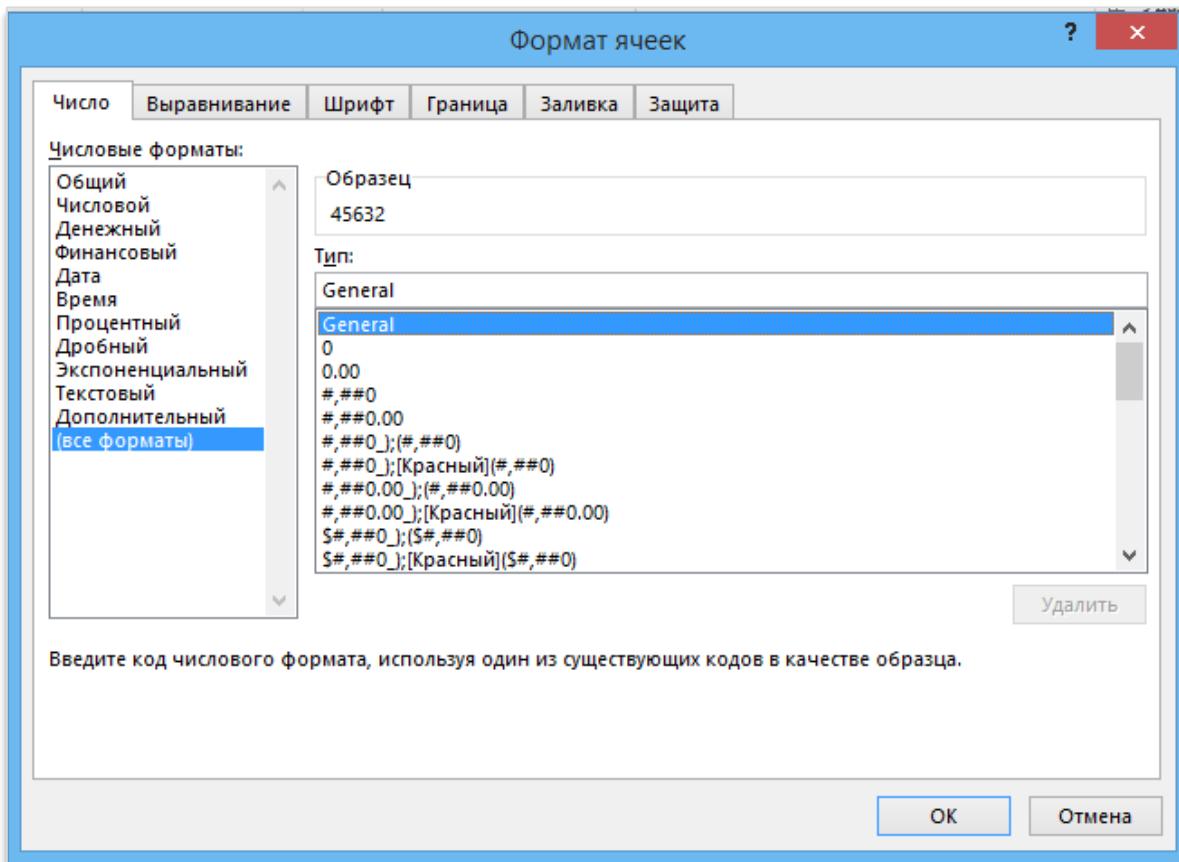


Рис. 2.5.75. Просмотр примеров форматов числа в Excel

Наиболее часто используемые форматы числа предлагаются в студии при наведении на поле для ввода формата.

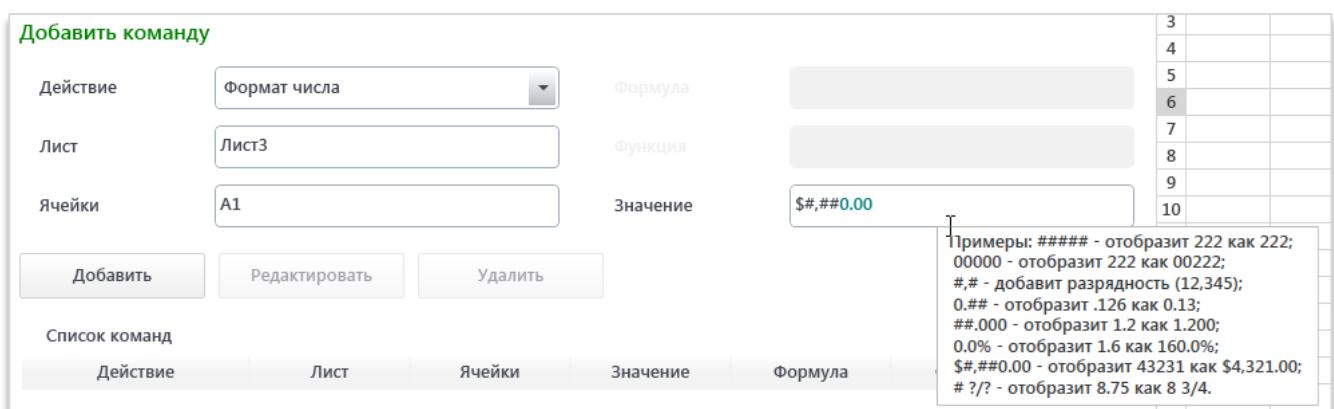


Рис. 2.5.76. Просмотр примеров форматов числа в студии

ПРИМЕР.

Отформатируем число, содержащееся в ячейке A1 в денежный долларовый формат. Для этого заполним действие так, как указано на Рис. 2.5.76.

Исходные данные представлены на Рис. 2.5.77:

	A	B
1	123456,00	
2	45632,00	
3	452,37	
4	12,45	
5		

Рис. 2.5.77. Пример исходных данных

Результат работы представлен на рисунке ниже:

	A
1	\$123 456,00
2	45632,00
3	452,37
4	12,45
5	

Рис. 2.5.78. Результат выполнения действия «Формат числа»

Действие «Выравнивание»

Действие применяет заданное выравнивание в указанных ячейках. В поле «Общее положение» предлагается выбрать одно из двух – по вертикали или по горизонтали применять выравнивание, выбранное в поле «Выравнивание содержимого», которое, в свою очередь, предлагает выбрать выравнивание содержимого по левому/правому краю, либо по верхнему/нижнему краю, либо по центру.

Добавить команду

Действие	Выравнивание	Общее положение	По вертикали
Лист	Лист1	Функция	
Ячейки	A1	Выравнивание содержимого	По центру

Рис. 2.5.79. Пример настройки действия «Выравнивание»

ПРИМЕР.

В заполненном на Рис. 2.5.79 действии настраивается выравнивание ячейки A1 по центру относительно вертикали.

Исходное положение текста в ячейке представлено на Рис. 2.5.80:

	A	B	C
1	Дата	Курс евро	Курс доллара
2			
3			

Рис. 2.5.80. Пример исходных данных

Результат работы представлен на Рис. 2.5.81:

	A	B	C
1	Дата	Курс евро	Курс доллара
2			
3			

Рис. 2.5.81. Результат выполнения действия «Выравнивание»

Действие «Перенос текста»

Действие добавляет в указанный диапазон надстройку ячеек «Перенос текста», функция которой – отображение текста ячейки в несколько строк, если он не помещается в одну.

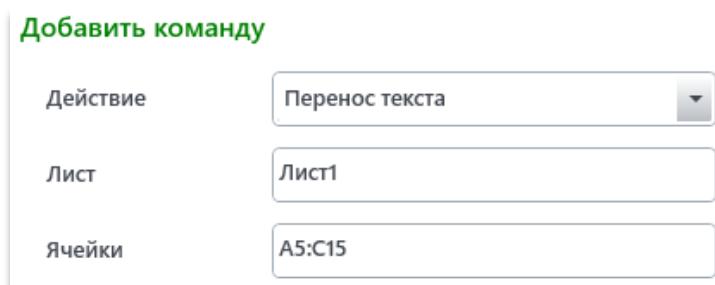


Рис. 2.5.82. Пример настройки действия «Перенос текста»

Действие «Шрифт»

Действие меняет шрифт, стиль или размер выбранных ячеек на указанном листе. Модуль предоставляет выбор всех стандартных шрифтов, доступных в пакете Microsoft Office.

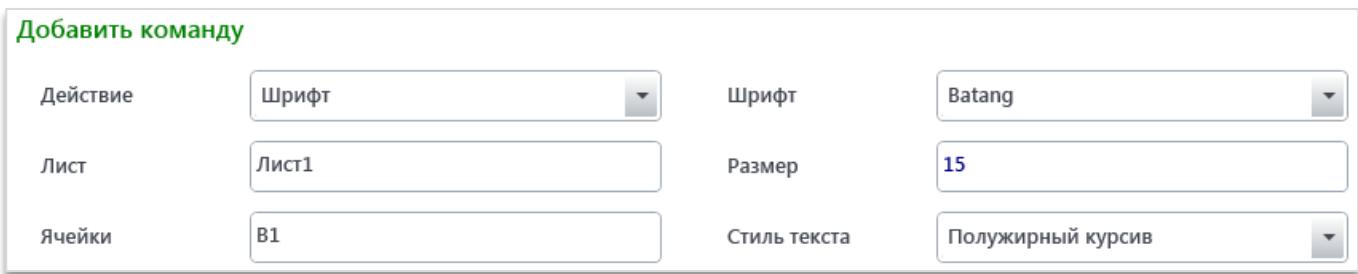


Рис. 2.5.83. Пример настройки действия «Изменение шрифта»

ПРИМЕР.

Изменим шрифт текста, находящегося в ячейке B1 так, как указано в действии на Рис. 2.5.83. Пример настройки действия «Изменение шрифта» – сам шрифт изменим на Batang, зададим стиль текста – полужирный курсив, а размер текста 16.

Исходное форматирование текста представлено на Рис. 2.5.84:

	A	B	C
1	Дата	Курс евро	Курс доллара
2			

Рис. 2.5.84. Пример исходных данных

Результат работы представлен на Рис. 2.5.85:

	A	B	C
1	Дата	<i>Курс евро</i>	Курс доллара
2			

Рис. 2.5.85. Результат выполнения действия «Шрифт»

Действие «Границы»

Действие добавляет или изменяет границы выбранных ячеек на указанном листе. Предполагает настройку расположения границ – внешние, внутренние, только правая и др., цвет границ и их стиль – толстая, тонкая, штриховая линия и др.

Добавить команду

Действие	Границы	Расположение	Внешние
Лист	Лист2	Стиль	Штриховой пунктир
Ячейки	A2:A6	Цвет	#000000 <input type="color"/>

Рис. 2.5.86. Пример настройки действия «Границы»

ПРИМЕР.

Добавим штрихпунктирную внешнюю границу для блока текста, находящегося в диапазоне A2:A6. Для этого необходимо заполнить действие так, как указано на Рис. 2.5.86.

Исходный вид блока текста представлен на Рис. 2.5.87:

	A	B
1		
2	Первый параметр	
3	Второй параметр	
4	Третий параметр	
5	Четвертый параметр	
6	Пятый параметр	
7		

Рис. 2.5.87. Пример исходных данных

Результат работы представлен на Рис. 2.5.88:

	A	B
1	-----	
2	Первый параметр	
3	Второй параметр	
4	Третий параметр	
5	Четвертый параметр	
6	Пятый параметр	
7		

Рис. 2.5.88. Результат выполнения действия «Границы»

Глава 6. Модуль «Закрыть Excel-файл»

Excel-модуль работает с файлами следующим образом – как только он начинает работу с некоторым файлом, то он открывает его (в фоновом, незаметном для пользователя, режиме), и держит открытым до конца работы робота. Это было сделано для ускорения работы модуля Excel – если модулей, использующих один и тот же файл, к примеру, несколько, пришлось бы несколько раз открывать и сохранять файл столько раз, сколько он используется что отразилось бы на быстродействии программы. В связи с этим, файл открывается при первом его использовании в модуле Excel и сохраняется только один раз – в самом конце. Но! Так как файл является открытым, с самим файлом нельзя ничего делать – переносить, архивировать, удалять за ненадобностью (например, провели расчеты в новом файле, взяли данные и затем его удалили), так как он занят. Для таких задач существует модуль «Закрыть Excel-файл» - он закрывает и сохраняет заданные ему файлы. Если эти файлы впоследствии снова будут использоваться – они опять откроются и, в следующий раз, будут закрыты либо этим же модулем, либо в конце работы робота.

Интерфейс модуля состоит из одного поля и списка путей к файлам, которые необходимо закрыть.

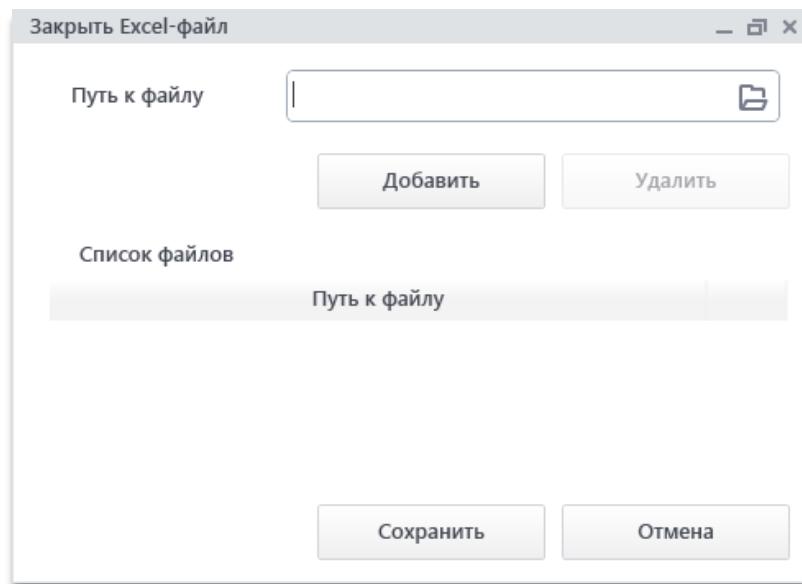


Рис. 2.6.1. Окно «Закрыть Excel-файл»

Поле «Путь к файлу» заполняется путем до файла, который необходимо закрыть. Он может содержаться в переменной, или можно выбрать его вручную, нажав на кнопку с изображением папки, находящуюся в поле для ввода.

По кнопке «Добавить» указанный в поле «Путь к файлу» путь будет занесен в «Список файлов».

По кнопке «Удалить» можно удалить выбранный путь из «Списка файлов».

ЧАСТЬ III. МОДУЛЬ «WORD»

Глава 7. Основные моменты модуля

Модуль Word обеспечивает работу с документами Microsoft Word пакета Microsoft Office. С помощью модуля можно создавать и редактировать документы с популярными текстовыми расширениями, такими как .doc, .docx, .html.

Интерфейс окна

Окно модуля состоит из трех основных частей, первая – часть, содержащая в себе поля для создания команд и кнопок управления ими, вторая – таблица «Список команд», которая будет содержать все добавленные действия для работы с файлом, и третья – окно с предварительным просмотром загруженного файла.

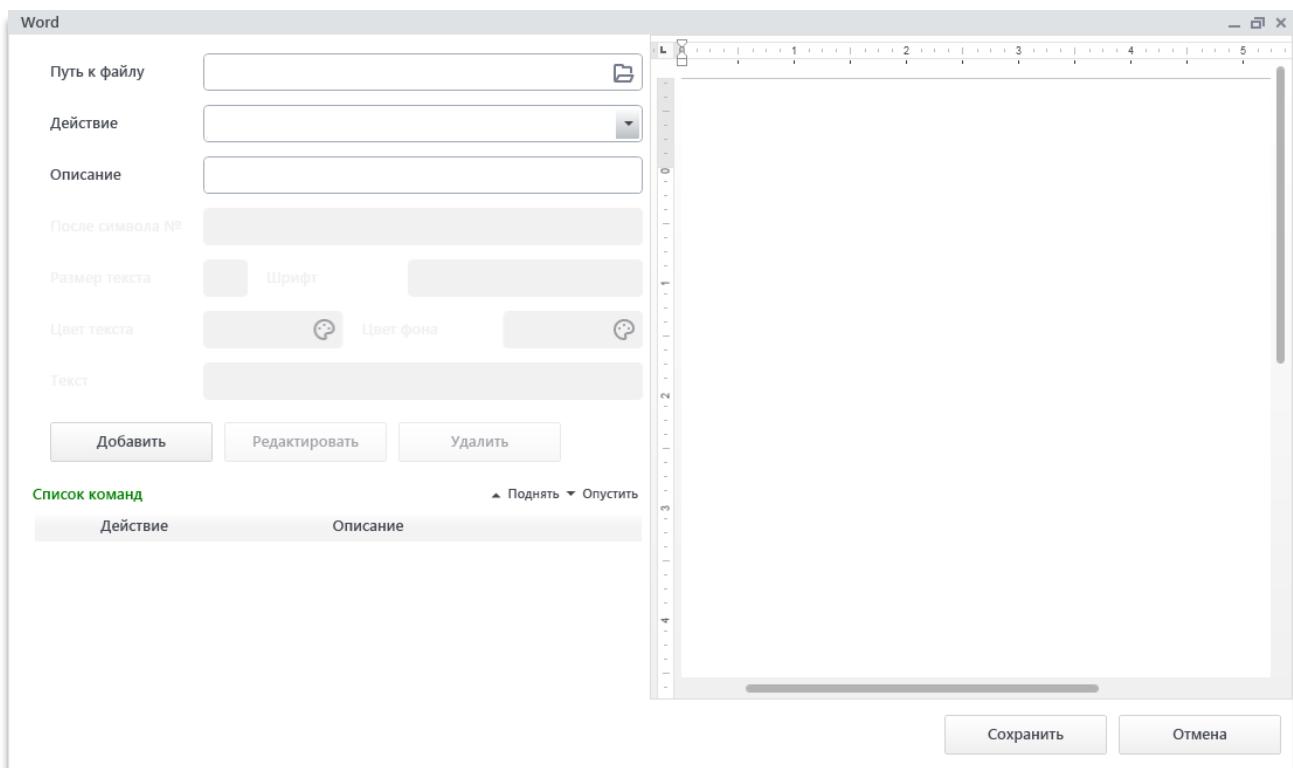


Рис. 3.7.1. Интерфейс окна

Окно предварительного просмотра позволяет просмотреть весь загруженный документ в том же виде, в котором он виден в самом Microsoft Word.

Раздел настройки модуля состоит из нескольких полей, доступность которых варьируется от необходимых для работы действий. Три первых поля являются основными и доступными всегда, причем обязательными являются только два из них:

- «Путь к файлу» - поле с кнопкой, открывающей диалог выбора файла. Данное поле предназначено для выбора существующего файла, с которым предстоит

далее работать, либо для введения в него пути, по которому необходимо будет создать новый файл. Обязательное поле;

- «Действие» - выпадающий список, состоящий из действий, которые можно применить к загруженному файлу или создать новый с указанным путем в поле «Путь к файлу». Обязательное поле;

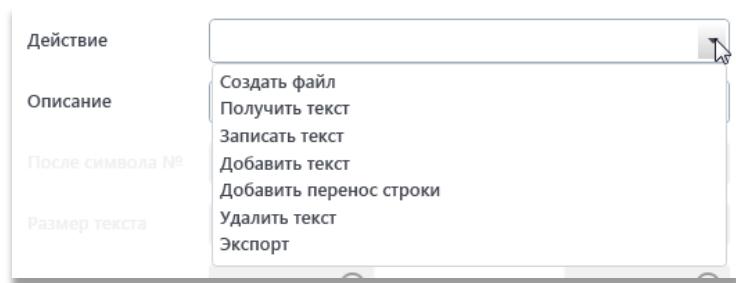


Рис. 3.7.2. Перечень доступных действий модуля

- «Описание» - поле, необходимое для удобной навигации по действиям, заполняется комментарием разработчика о создаваемом действии. Необязательное поле.

Все возможные действия будут описаны в следующей главе.

Для добавления действия в таблицу «Список команд» необходимо воспользоваться кнопкой «Добавить» под полями настройки действия.

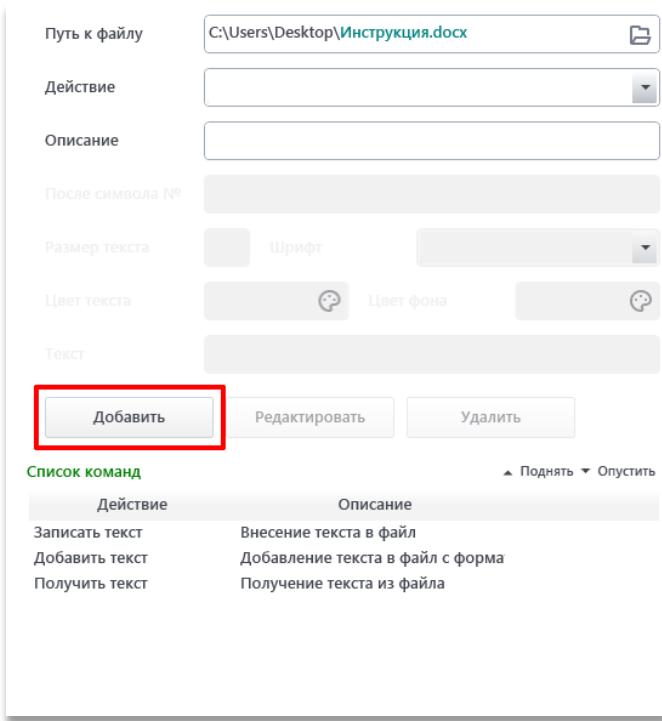


Рис. 3.7.3. Расположение кнопки «Добавить»

После нажатия на кнопку созданное действие добавится в «Список команд». Для редактирования существующей команды или её удаления, необходимо выбрать строку с командой в таблице команд и нажать «Удалить» для её удаления, или отредактировать необходимые поля действия и затем нажать «Редактировать».

Для того, чтобы поменять существующие действия местами, необходимо выбрать одно действие и нажать на необходимые кнопки «Поднять» или «Опустить», которые расположены в правом углу над таблицей с командами.

Список команд		▲ Поднять ▾ Опустить
Действие	Описание	
Записать текст	Внесение текста в файл	
Добавить текст	Добавление текста в файл с форм	
Получить текст	Получение текста из файла	
Записать текст	Внесение текста в файл	

Рис. 3.7.4. Кнопки «Поднять/Oпустить» и выбор строки в таблице

Глава 8. Действия модуля

Создать файл

Данное действие создает файл с указанным путем, именем и расширением. Для настройки действия необходимо заполнить поле «Путь к файлу» строкой типа «*Путь до файла/Название файла.Требуемое расширение*», например, «C:/Reports/Documents/Test.docx».

Путь к файлу	C:/Reports/Documents/Test.docx
Действие	Создать файл
Описание	Создание нового файла Text.docs

Рис. 3.8.1. Пример заполнения действия «Создать файл»

Получить текст

Действие «Получить текст» позволяет считать весь текст из загруженного файла в переменную. Для настройки действия появляется еще одно поле помимо стандартных трёх – «Переменная», в которую записывается название переменной (с символом «v.»), в которую будет записан считанный текст.

Путь к файлу	C:\Users\Desktop\Инструкция.docx
Действие	Получить текст
Описание	Получение текста из файла
После символа №	
Размер текста	Шрифт
Цвет текста	Цвет фона
Переменная	v.text

Рис. 3.8.2. Пример настройки действия «Получить текст»

Записать текст

С помощью действия можно записать имеющийся текст, например, считанный из другого текстового файла, в текущий загруженный файл. Для настройки обязательно заполнение поля «Значение», куда вводится либо переменная (с символом «v.»), содержащая текст, либо сам текст. Обратите внимание, что текст, введенный данным действием, полностью затрет существующий текст в файле.

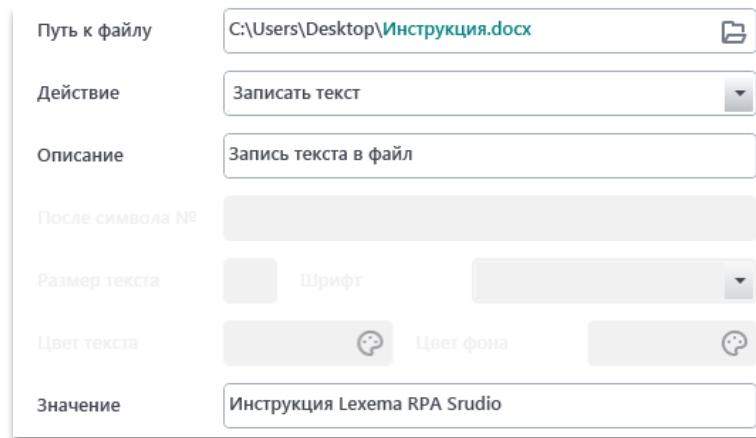


Рис. 3.8.3. Пример настройки действия «Записать текст»

Добавить текст

Данное действие позволяет добавлять текст к существующему с применением необходимого форматирования. Для редактирования открываются все поля модуля. Далее будет подробно описано каждое поле, отличное от основных трёх.

«После символа №» - данное поле ожидает ввод целого числа, обозначающего, после какого по счету символа необходимо вставить текст (с учетом пробелом и символов переноса строки или разрыва страницы).

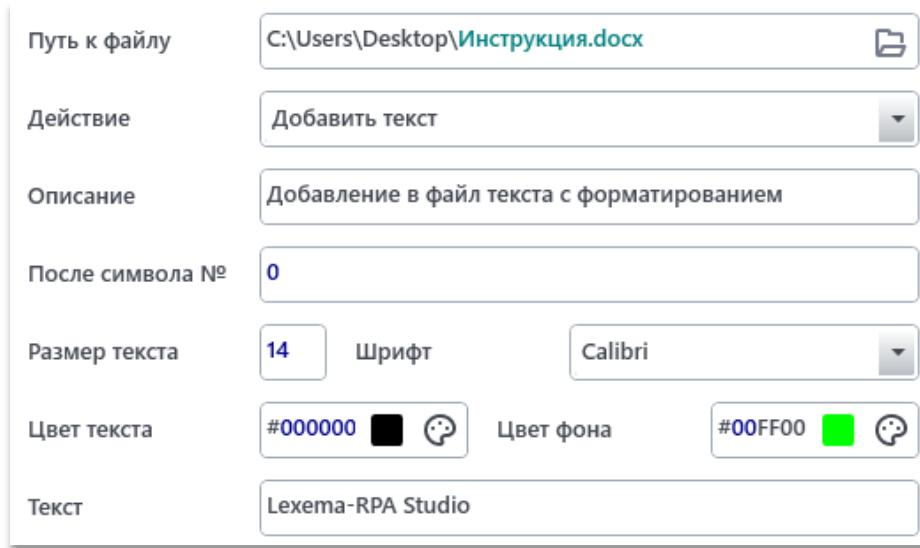


Рис. 3.8.4. Пример настройки действия «Добавить текст»

«Размер текста» - аналогичен полю «Размер шрифта» в Microsoft Word – устанавливает размер добавляемого текста. Ожидается ввод целого или десятичного числа.

«Шрифт» - выпадающий список, содержащий в себе набор стандартных шрифтов пакета Microsoft Office – устанавливает шрифт добавляемого текста.

«Цвет текста» и «Цвет фона» – поля с иконкой палетки сбоку, по нажатию на которые открывается диалог выбора цвета – устанавливают цвет добавляемого текста и его фона соответственно. Вводится в формате цвета HEX – шестнадцатеричное представление RGB.

«Текст» - поле, в которое указывается добавляемый текст или переменная, которая его содержит.

Добавить перенос строки

Данное действие добавляет символ переноса строки в файл. Не требует заполнения дополнительных полей.

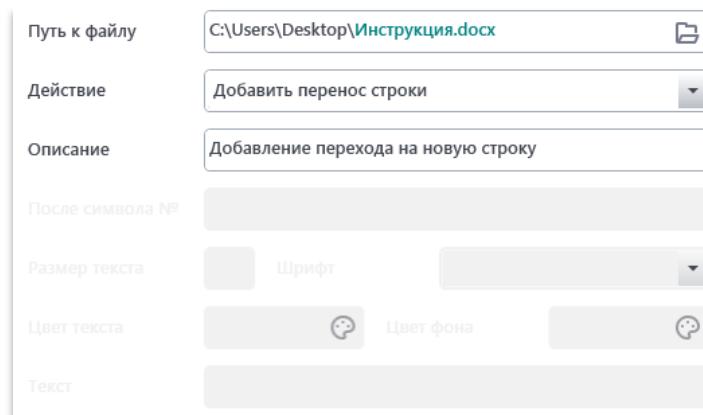


Рис. 3.8.5. Пример настройки действия добавление перехода на новую строку

Удалить текст

С помощью данного действия можно удалить текст, зная количество символов в нем и его расположение. Для настройки необходимо ввести в поле «После символа №, количество» два целых числа через запятую, где первое число – номер символа, после которого необходимо удалить текст, а второе – количество символов в удаляемом тексте.

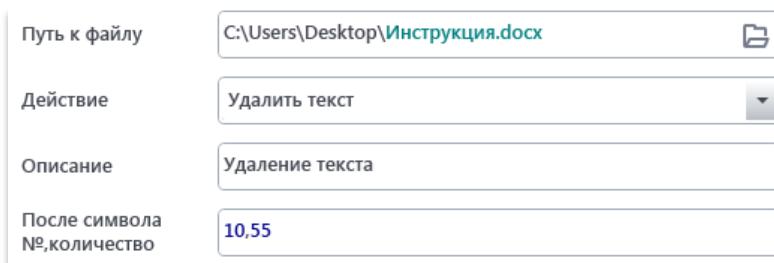


Рис. 3.8.6. Пример настройки действия «Удалить текст»

Экспорт

С помощью данного действия можно менять расширение загруженного файла. Для настройки действия нужно выбрать из выпадающего списка «Тип» тип файла, в который нужно экспортить, и указать путь с названием и расширением будущего файла в поле «Путь выходного файла».

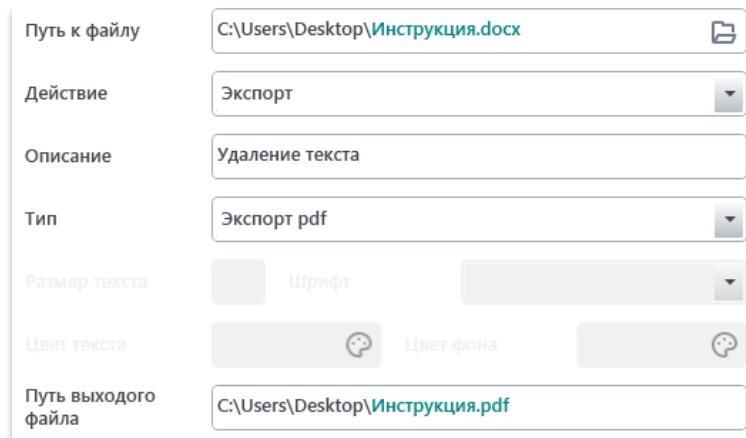


Рис. 3.8.7. Пример настройки действия «Экспорт»

ЧАСТЬ IV. МОДУЛЬ «КЛИКЕР» И МОДУЛИ ПОИСКА ИЗОБРАЖЕНИЙ

Глава 9. Модуль «Кликер»

Модуль «Кликер» предназначен для записи последовательности действий, выполняемых пользователем, а именно кликов по определенным местам экрана и нажатий кнопок на клавиатуре или мыши. С помощью этого модуля можно производить интеграции между самыми разными программами, так как он точно имитирует действия пользователя, то есть, если Вы сможете это сделать, то это сможет сделать и «Кликер».

Использование роботов-кликеров, которые ищут координаты или изображения, рекомендуется на компьютерах с таким же разрешением, как и тот, на котором был написан данный робот.

Создание макроса

При открытии модуля «Кликер» появится диалоговое окно «Запись действий макроса», состоящее из трех элементов – кнопки «Начать запись», области, в которой будут отображены записанные действия, и кнопки «Сохранить».

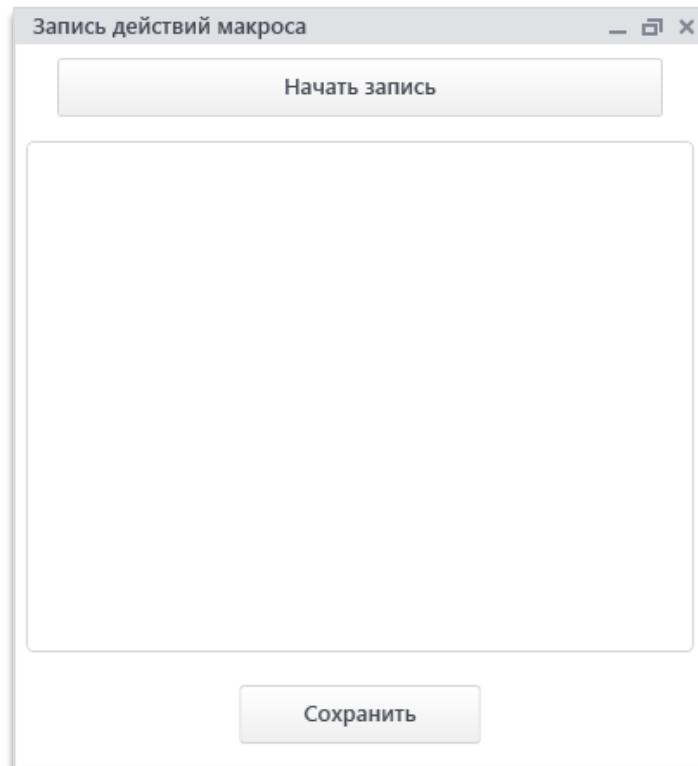


Рис. 4.9.1. Запись действий макроса

По нажатию кнопки «Начать запись» программа переходит в режим **ожидания нажатия кнопки «TAB»**, по которой открывается окно создания действия (команды).

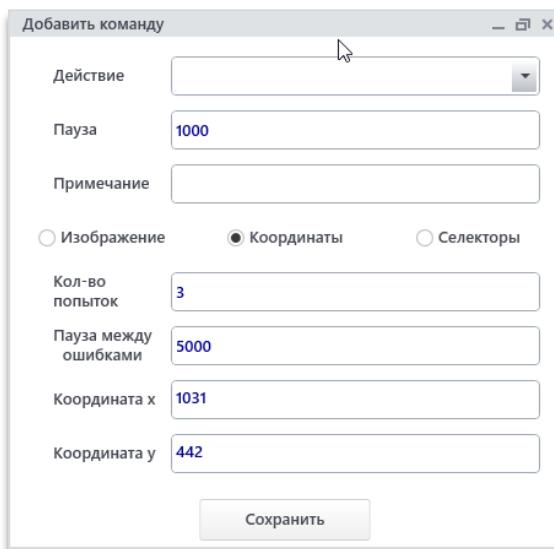


Рис. 4.9.2. Создание команды

В открывшемся окне происходит настройка действия (описание всех возможных действий будет дано ниже). По завершению создания команды и нажатию кнопки сохранить, новое действие будет добавлено в макрос.

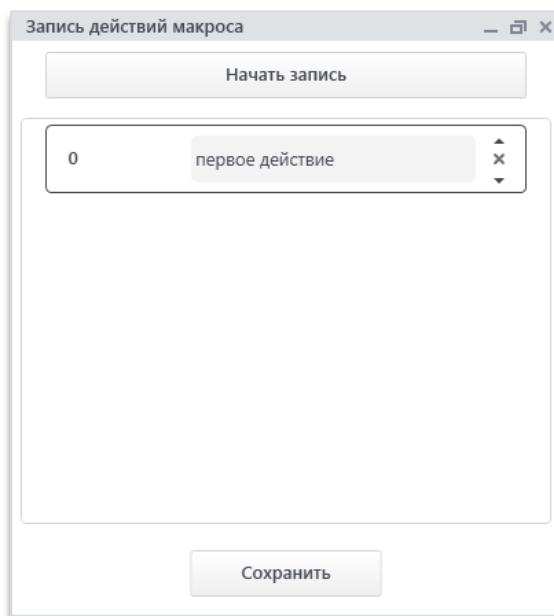


Рис. 4.9.3. Добавление команды

Изменить положение действия в списке команд можно с помощью кнопок с изображением стрелок, расположенных в правой части блока.

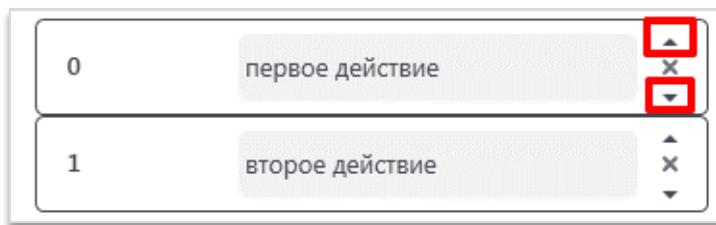


Рис. 4.9.4. Изменение порядка команды

Кнопка с изображением крестика, расположенная между стрелками, предназначена для удаления действия из списка команд.

Для сохранения макроса необходимо нажать на кнопку «Сохранить».

Создание команды

Окно создания команды состоит из статичных и изменяющихся полей. Статичными являются такие поля, как «Действие», «Пауза» и «Примечание».

Статичные поля

Действие

Поле «Действие» состоит из перечня доступных действий для записи макроса – нажатие кнопок мыши и клавиатуры, наиболее часто встречающиеся сочетания кнопок (скопировать, вставить). Далее будут приведены все действия, доступные в модуле.

Пауза

Поле «Пауза» предназначено для создания эффекта ожидания после выполняемой команды. Значение указывается в миллисекундах.

Например, после нажатия на иконку браузера работу следует подождать несколько секунд, пока браузер запустится и будет готов к работе – в данном примере в команде открытия браузера (кликнуть мышкой по его иконке) в поле «Пауза» следует написать некоторое значение, например, 5000, т.е. 5 секунд.

Примечание

Поле «Примечание» позволяет писать комментарии, чтобы не запутаться в командах.

Поля, появляющиеся только при определенных действиях

Переключатели «Изображение», «Координаты» и «Селекторы»

При создании действий, связанных с кликом или наведением мыши, вам предлагается выбрать, куда кликать – по определенным координатам или по изображению. Если программа, с которой вы работаете, статичная и не меняет расположение своих кнопок, то можно воспользоваться кликом по координатам. Тогда возникает вопрос, как рассчитать координаты?

Все очень просто – при нажатии кнопки «ТАВ» нужно располагать мышь ровно в том месте, куда следует нажать. В диалоговом окне создания команды координаты уже будут заполнены.

Если программа может менять расположение её элементов, то настоятельно рекомендуем использовать клик по изображению. Для этого нажмите на переключатель «Изображение».

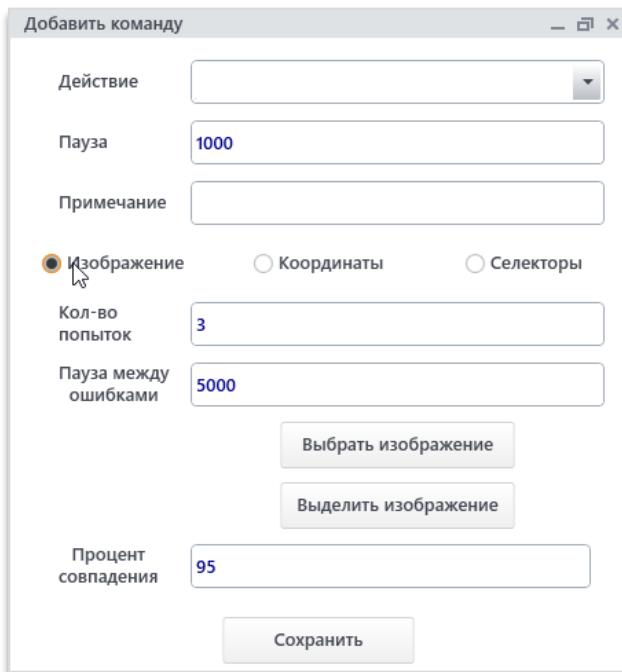


Рис. 4.9.5. Создание команды

При изменении переключателя изменится нижняя часть окна – появятся кнопки «Выбрать изображение», «Выделить изображение» и поле для ввода «Процент совпадения».

Изображение, по которому потребуется кликнуть для выполнения действия, можно настроить двумя способами:

1) загрузить имеющееся изображение, нажав на кнопку «Выбрать изображение»

2) выбрать фрагмент снимка окна, в котором будет воспроизводиться действие, с помощью кнопки «Выделить изображение». После нажатия на кнопку студия сделает снимок экрана и предложит вам выбрать фрагмент получившегося изображения.

При использовании клика по изображению, *обратите внимание* на то, что, если на экране таких изображений будет несколько, действие будет применено к *первому* найденному!

Поле «Процент совпадения» необходимо для случаев, когда искомое изображение может немного отличаться от того, что может быть в реальности (например, наличие выделения вокруг кнопки, если она выбиралась ранее, и отсутствие такого выделения). В подобных случаях уменьшайте процент совпадения до тех пор, пока изображение не будет находиться и в том, и в другом случае. Однако если процент совпадения уменьшить слишком сильно, модуль может найти то, что не планировали искать.



Рис. 4.9.6. Пример отличия искомого изображения от действительного

Поля «Количество попыток» и «Пауза между ошибками» предназначены для случаев, когда программа не смогла найти заданное ей изображение. Вы можете указать, сколько раз следует пытаться искать требуемое изображение и сколько секунд ждать между попытками. Например, после запуска браузера нужно нажать на кнопку с иконкой «Плюсик» – создание новой вкладки. Если браузер будет загружаться слишком долго, больше 5-ти секунд (это время указано в паузе команды открытия браузера), тогда программа начнёт искать «плюсик», которого еще нет. Затем программа подождет указанное ей количество секунд между попытками и попробует снова. Как только изображение будет найдено, программа подождет то количество секунд, которое указано в поле «Пауза» данной команды, и перейдет к следующей.

Последний вариант поиска места для наведения мыши – «Селекторы». Этот вариант считается самым надежным, но поддерживается узким кругом программ. Селекторы Windows позволяют находить путь к элементу на экране используя не просто координаты или изображение, а путь относительно родительского компонента экрана с использованием специальной разметки. Они работают с системой компьютера, некоторыми desktop-приложениями. Использование подобно работе с координатами – необходимо навестить на искомый элемент и вызвать окно настройки действия по кнопке «ТАБ». При переключении настроек на «Селекторы», путь до элемента уже будет введен.

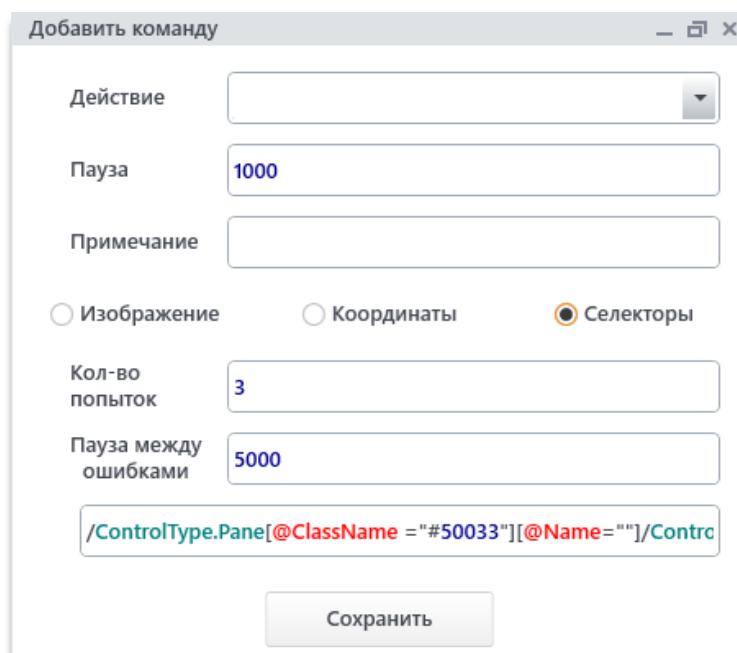


Рис. 4.9.7. Рис. 4.9.8. Создание команды через селекторы Windows

Прокрутка мыши

Данное поле появляется при действии «Прокрутка мыши». В него вводится значение в пикселях, на сколько необходимо прокрутить колесико мыши. Если прокрутка вверх, то значение должно быть положительным, если вниз, то отрицательным.

Во всех полях окна создания команды **возможно использование переменных**.

Описание действий, доступных в модуле

Действие «Кликнуть мышкой»

Основное действие модуля – нажатие левой кнопкой мыши по указанным координатам или изображению.

Действие «Кликнуть мышкой дважды»

Двойной клик по указанным координатам или изображению, например, для открытия приложения с рабочего стола.

Действие «Кликнуть правой кнопкой мыши»

Клик правой кнопкой мыши, например, по приложению на рабочем столе, чтобы открыть его контекстное меню.

Действие «Кликнуть колесиком мыши»

В некоторых программах есть возможность использовать клик колесика мыши для исполнения некоторых действий, например, закрыть вкладку в браузере.

Действие «Прокрутка мыши»

Прокрутка колесом мыши вниз или вверх по странице. Здесь координаты играют роль того, где будет находиться курсор при прокрутке колеса, так как ползунков на экране может быть несколько. Для прокрутки экрана вниз в поле «Прокрутка мыши» должно быть введено отрицательное значение (-150), вверх - положительное (150).

ПРИМЕР.

Для прокрутки вниз конкретного ползунка необходимо навести на него курсор, нажать клавишу «ТАВ» на клавиатуре, выбрать действие «прокрутка мыши» и заполнить поле «Прокрутка мыши», как показано на Рис. 4.9.9:

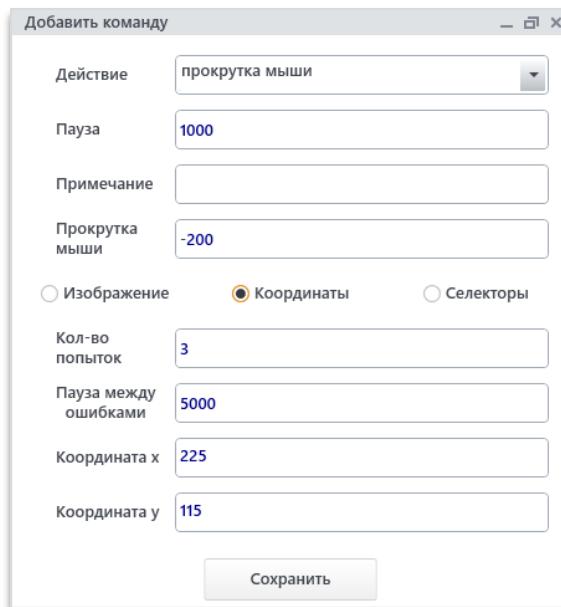


Рис. 4.9.9. Пример настройки команды

Действие «Ввести текст»

Данное действие вводит текст, который записан в поле «Текст». Поле, куда вводится текст, должно быть выделено (должен быть фокус, стоять каретка).

В некоторых программах необходим специфичный ввод текста – в конкретной кодировке. В подобных случаях следует выбрать кодировку из предложенных в выпадающем списке «Использовать кодировку», или указать свою. Текст может лежать в переменной и быть скопированным из какой-либо программы, использующей конкретную кодировку. В таком случае, при вставке скопированного текста в другую программу, необходимо сначала считать его в кодировке исходной программы. Кодировка исходной программы вводится или выбирается в поле «Начальная кодировка». Заполнение этих полей **необязательно**.

Флажок «Использовать WinInput» позволяет включить режим более корректного и быстрого ввода текста, но он может не поддерживаться некоторыми программами. Его использование **необязательно**.

Пример заполнения действия:

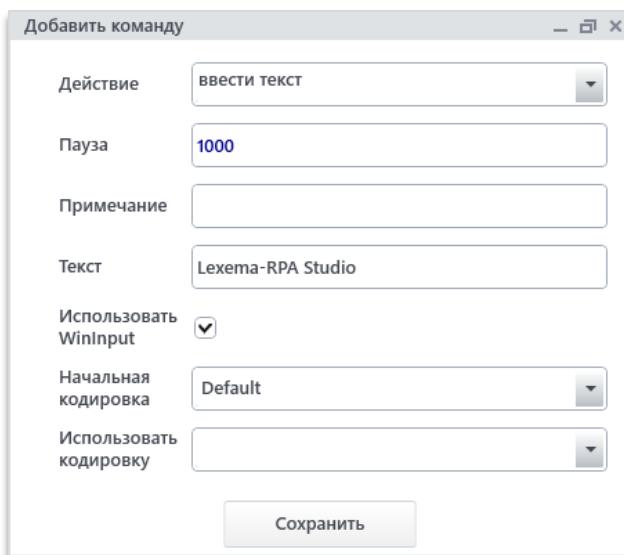


Рис. 4.9.10. Пример заполнения

Действие «Нажать кнопку»

Имитирует нажатие клавиши клавиатуры. Необходимую клавишу следует выбрать в выпадающем списке «Кнопка». В списке предложены все возможные клавиши, которые могут присутствовать как на простых, так и на расширенных клавиатурах (изменение громкости звука, запуск/остановка музыки и другие нестандартные кнопки).

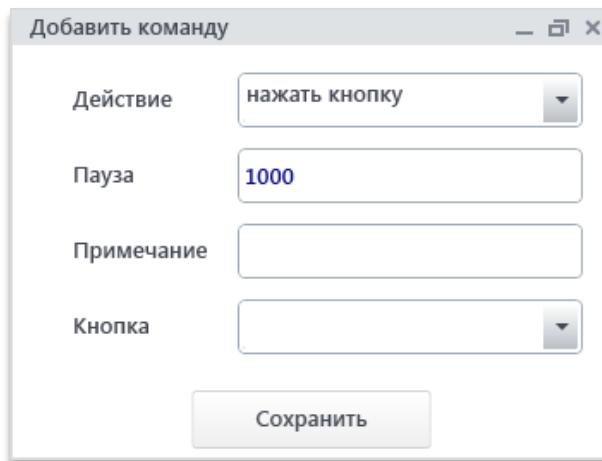


Рис.4.9.11. Действие «Нажать кнопку»

Действие «Нажать две кнопки»

Имитирует нажатие сочетания двух клавиш клавиатуры, например CTRL+A – выделение всего текста в поле. Необходимые кнопки выбираются в двух выпадающих списках «Кнопка». Список кнопок тот же, что и в действии «Нажать кнопку».

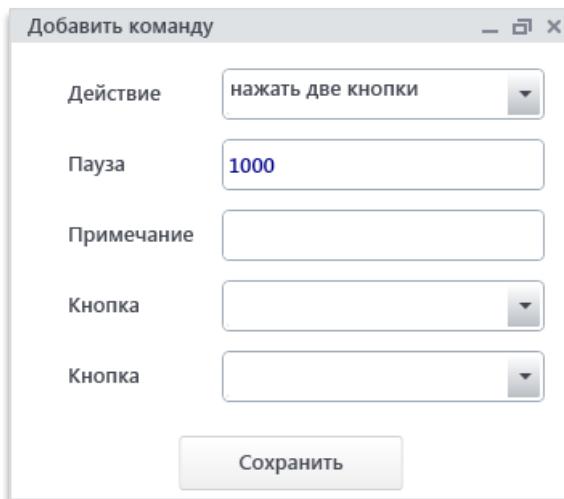


Рис. 4.9.12. Действие «Нажать две кнопку»

Действие «Скопировать»

Имитирует нажатие сочетания клавиш клавиатуры **CTRL+C**. Копирует выделенное заранее значение в буфер обмена и записывает его, при необходимости, в переменную, указанную в поле «Записать в». Данное поле **не обязательно** к заполнению. Ввод переменной начинается с символов «v.».

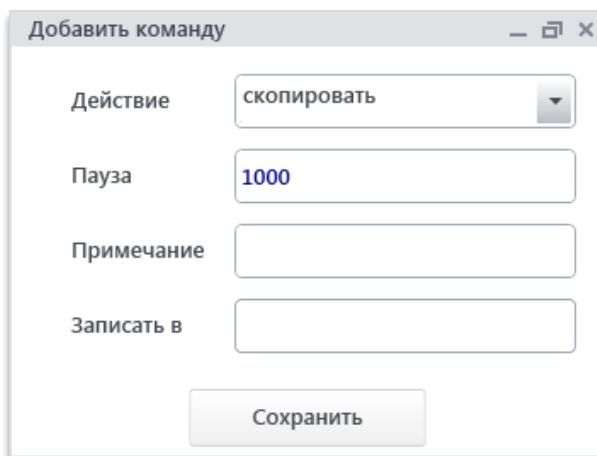


Рис. 4.9.13. Действие «Скопировать»

Действие «Вставить»

Имитирует нажатие сочетания клавиш клавиатуры **CTRL+V**. Вставляет в выбранное поле значение из буфера обмена.

Действие «Пока не пропадет изображение»

Ожидает отсутствия заданного изображения на экране в течение времени, дающегося на обнаружение ошибки (значение поля «Кол-во попыток» умноженное на значение поля «Пауза между ошибками»).

Глава 10. Модуль «Поиск изображения»

Модуль «Поиск изображения», в основном, используется в связке с модулем «Кликер». Его задача – определить, существует ли на текущем экране заданное изображение, и если да, то вернуть координаты его центра.

Интерфейс модуля

Окно модуля выглядит следующим образом:

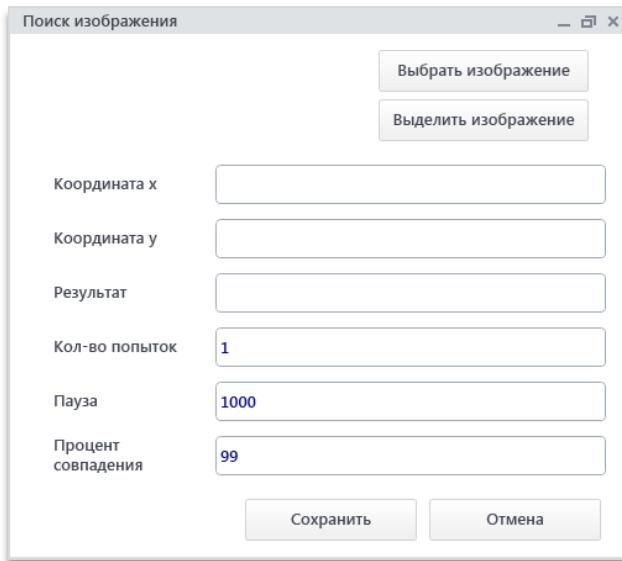


Рис. 4.10.1. Окно модуля

Кнопки «Выбрать изображение» и «Выделить изображение» позволяют выбрать изображение для поиска.

Поле «Координата x». В поле вводится *переменная* (начиная с символов «v.»), в которую будет записана координата x искомого изображения, если оно найдется.

Поле «Координата y». Аналогично полю «Координата x», в данном поле указывается переменная, в которую будет записана координата y.

Поле «Результат». В поле вводится название переменной, в которой будет итоговый результат поиска изображения (после использования всех попыток его найти). Результатирующее значение будет представлено в виде строки: «true» - найдено, «false» - не найдено.

Поля «Кол-во попыток», «Пауза» и «Процент совпадения» аналогичны полям модуля «Кликер».

Поле «Кол-во попыток». В поле указывается количество попыток поиска изображения.

Поле «Пауза» предназначено для указания паузы между попытками поиска в миллисекундах.

Поле «Процент совпадения» - насколько сильно совпадает изображение, которое будет искалься, с фрагментом изображения на текущем экране. Вводится число от 0 до 100, чем больше, тем больше должно быть сходства между изображениями.

Глава 11. Модуль «Поиск изображений»

Модуль «Поиск изображений» аналогичен модулю «Поиск изображения» с одним лишь отличием в том, что этот модуль возвращает список всех найденных изображений, а не одного.

Интерфейс модуля

Интерфейс этого модуля очень похож на интерфейс модуля «Поиск изображения», но в нем вместо полей «Координата X», «Координата Y» и «Результат» только одно поле – «Переменная», а также появилось поле «Минимальная дистанция».

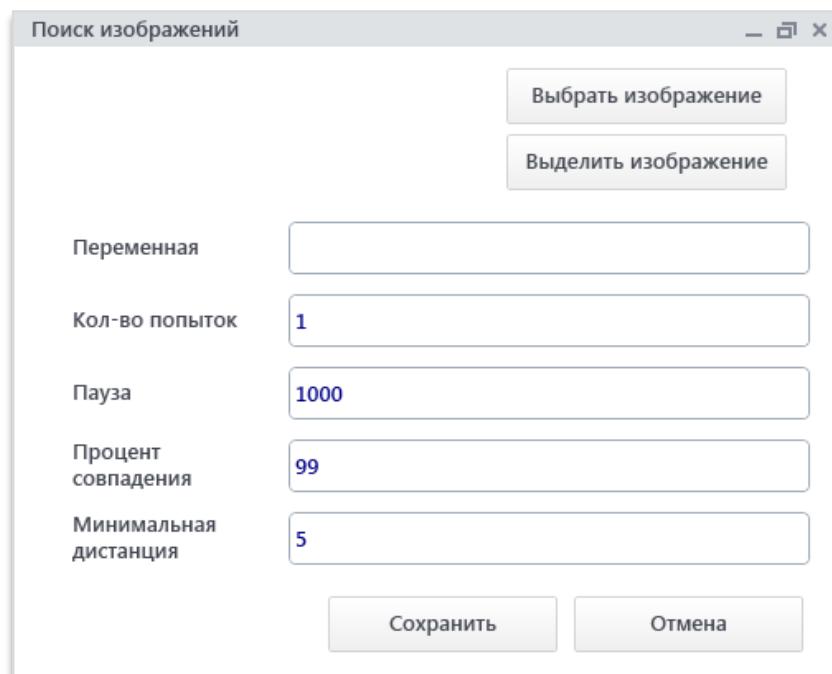


Рис. 4.11.1. Окно модуля

В поле «Переменная» вводится название переменной с символом «v.», в которую запишется результат поиска.

Поле «Кол-во попыток» заполняется числом, указывающим, сколько раз пробовать искать изображения, если с первого раза ни одного найти не удалось.

Поле «Пауза» предназначено для указания паузы между попытками поиска в миллисекундах.

В поле «Процент совпадения» вводится значение от 0 до 100, указывающее, насколько сильно совпадают изображение, которое будет искаться, с фрагментом изображения на текущем экране. Чем больше число, тем сильнее они должны совпадать.

Поле «Минимальная дистанция» заполняется числом, указывающим, какое минимальное количество пикселей должно быть между двумя одинаковыми изображениями. В некоторых задачах может быть потребоваться отсекать изображения, которые расположены достаточно близко друг к другу и получать только первое из этих изображений (начиная с

верхнего левого угла). **Обратите внимание**, если это расстояние поставить совсем маленьким (не менее 5 пикселей) или равным 0, то одно изображение может быть найдено несколько раз!

Выходные данные

В указанную переменную будет записан список объектов, количество которых будет равно количеству найденных изображений на экране. Каждый объект соответствует найденному изображению. Для обращения к конкретному изображению списка необходимо указывать его номер (нумерация с 0) в квадратных скобках после названия переменной, например `v.listImages[0]`.

Каждый объект содержит три поля: «`x`», «`y`» и «`similarity`». В поле «`x`» содержится координата x выбранного изображения, в поле «`y`» - координата y, а в поле «`similarity`» - процент совпадения найденного изображения с искомым.

ПРИМЕР.

Пусть мы хотим с помощью робота найти количество одинаковых изображений, например, непрочитанных писем с Яндекс-почты, которые выделены небольшим синим кружочком около письма:

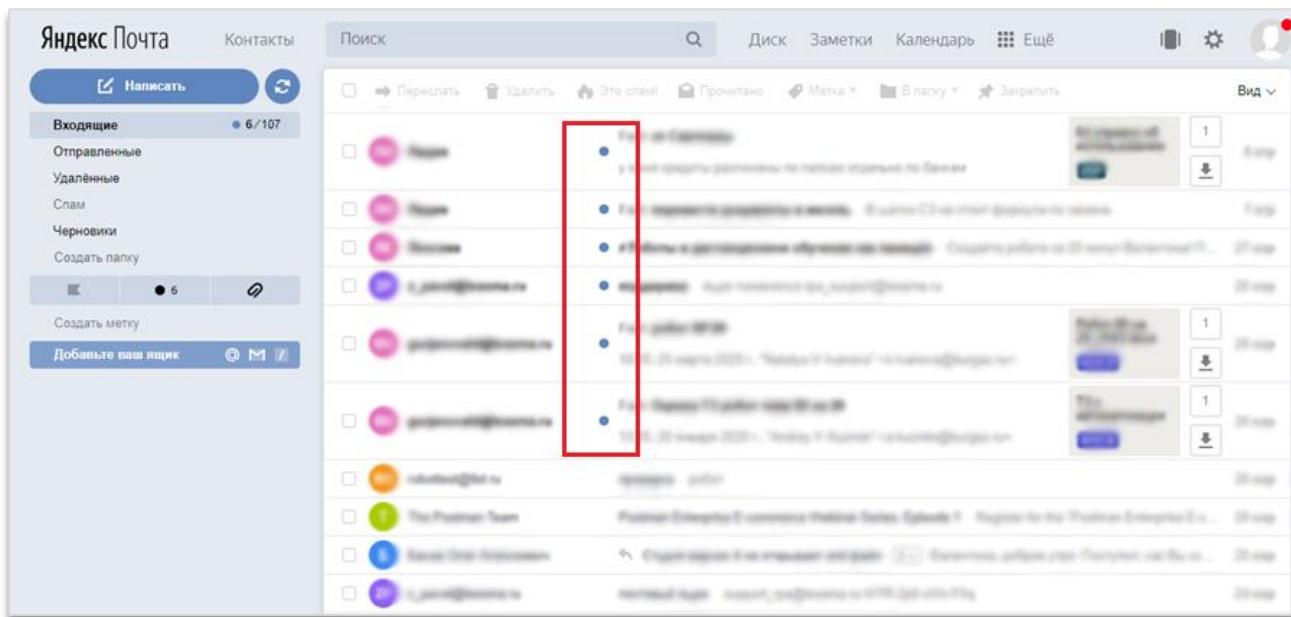


Рис. 4.11.2. Пример

Для этого создадим действие «Поиск изображений» и выберем в нем в качестве искомого изображения эту синенькую точку:

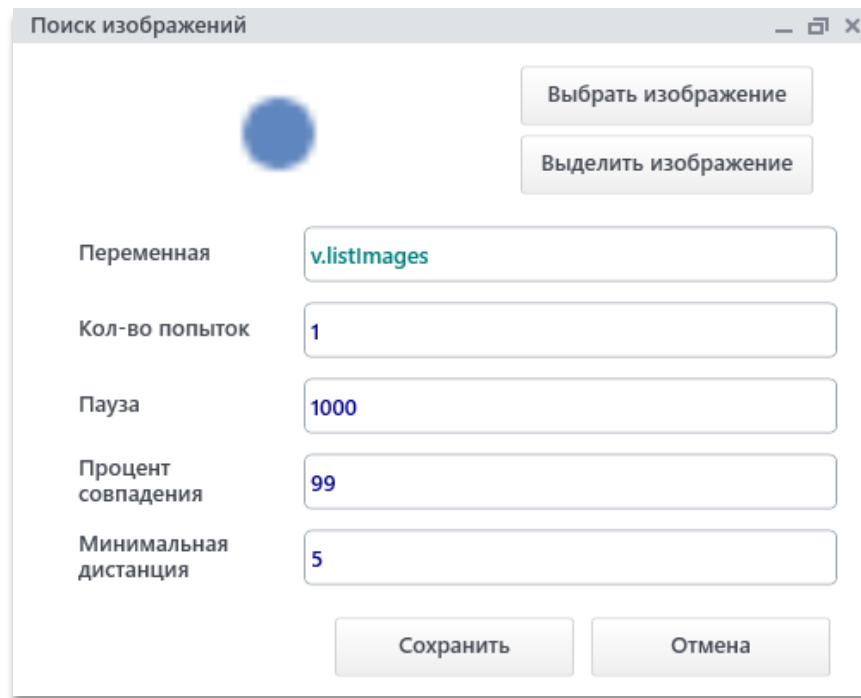


Рис. 4.11.3. Настройка действия

Для нажатия на каждую такую точку и прочтения сообщений, необходимо добавить цикл по количеству изображений, и в каждом шаге цикла нажимать на координаты v.listImages[v.i].x и v.listImages[v.i].y, где v.i – текущий шаг цикла.

Для определения количества элементов массива можно воспользоваться методом языка JS – v.listImages.length.

ЧАСТЬ V. МОДУЛЬ «РАБОТА С ФАЙЛАМИ»

Глава 12. Основные моменты модуля

Модуль «Работа с файлами» предназначен для изменения расположения файлов, их копирования, архивирования, создания папки и переноса файлов в неё, в общем, для манипуляции с файлами и директориями.

Интерфейс модуля

Окно модуля объединяет три группы элементов.

Первая группа элементов – это набор полей и кнопок для создания команды. Вторая – непосредственно список команд. И третья – это область предварительного просмотра текстовых файлов.

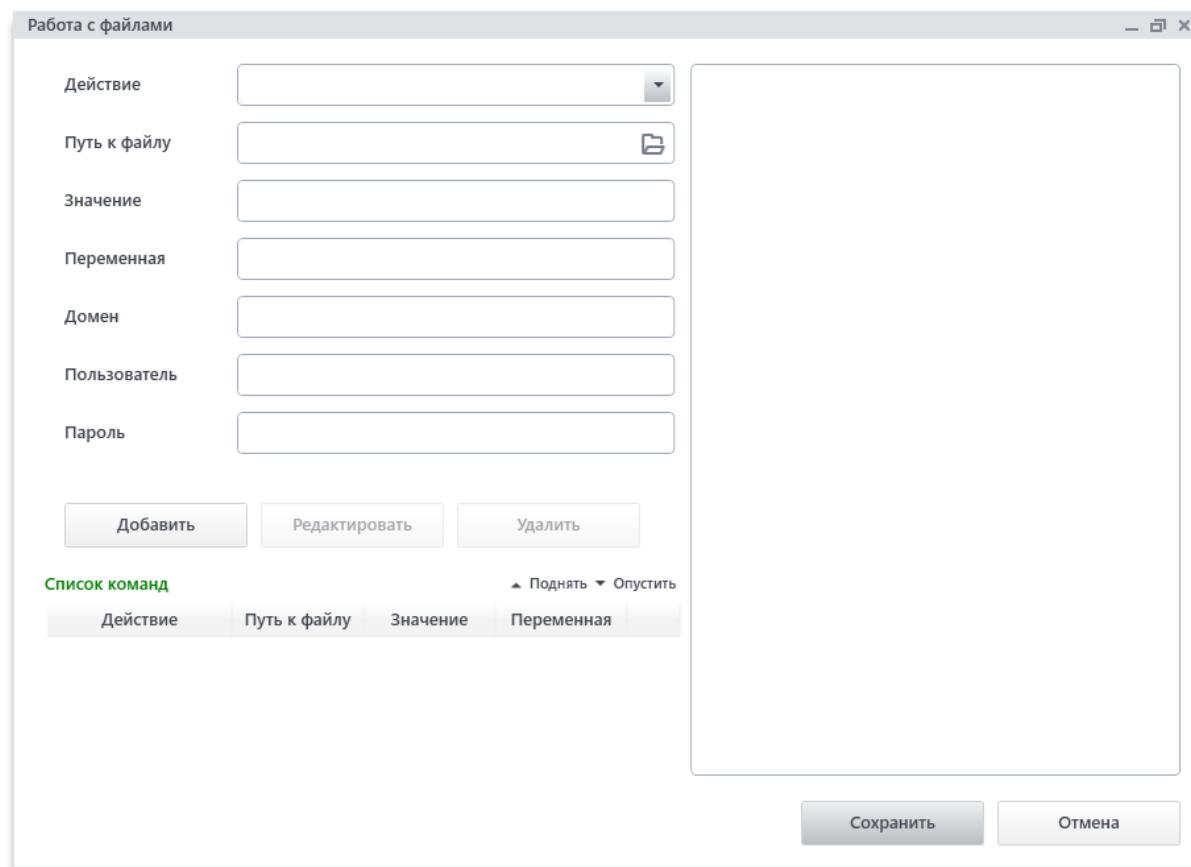


Рис. 5.12.1. Окно модуля

Набор доступных полей для заполнения меняется в зависимости от выбранного действия, подробное описание всех полей будет дано в Глава 13.

Функциональность модуля

Создание команды

Команда создается следующим образом – выбирается необходимое действие, заполняются остальные поля – нажимается кнопка «Добавить», по которой команда добавляется в «Список команд».

Редактирование команды

Для того чтобы отредактировать команду, выделите её в списке команд щелчком левой кнопкой мыши, после чего все поля в левой верхней части окна «Работа с файлами» заполняются в соответствии с выбранной командой. Измените те поля, которые требуется отредактировать, и, убедившись, что в списке команд все еще выбрано нужное действие, нажмите на кнопку «Редактировать».

Удаление команды

Для удаления команды необходимо выбрать её в списке команд и нажать на кнопку «Удалить».

Отключение/включение команды

Созданные команды можно отключать и обратно включать. Отключенные команды не будут выполняться. Сделать это можно, нажав на соответствующую отключаемой команде строку правой кнопкой мыши и выбрав «Отключить/включить».

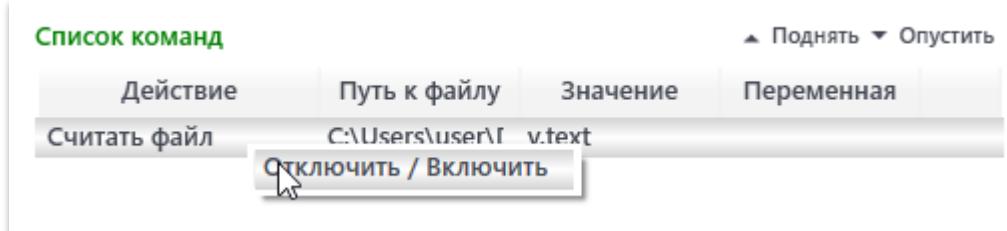


Рис. 5.12.2. Конекстное меню команды

Выбор файла/папки

Кнопка в заполняемых полях открывает стандартный диалог выбора файла или папки, в зависимости от выбранного действия.

Глава 13. Описание действий, доступных в модуле

Модуль содержит 19 действий, позволяющих работать с папками и файлами, каждое из которых состоит из своего набора полей для настройки. Ниже описаны все действия модуля.

Модуль имеет возможность работы с сетевыми папками, для этого предусмотрены поля «Домен», «Пользователь» и «Пароль». Остальные поля заполняются в соответствии с выбранным действием.

Перечень и описание действий

Действие «Сохранить в файл»

Действие перезаписывает существующий текстовый файл, записывая в него строку, указанную в поле «Значение».

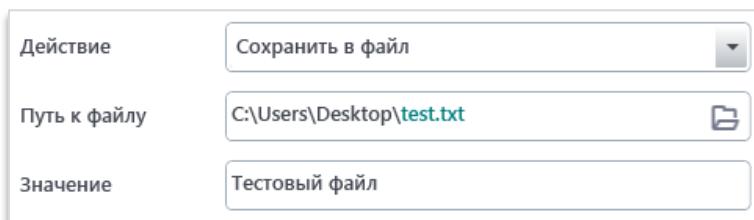


Рис. 5.13.1. Пример настройки действия «Сохранить в файл»

Действие «Считать файл»

Действие считывает весь текст из выбранного файла и записывает его в указанную переменную.

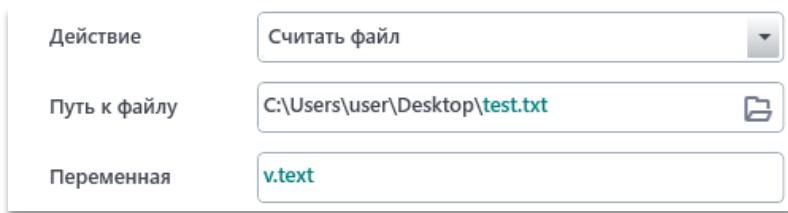


Рис. 5.13.2. Пример настройки действия «Считать файл»

Действие «Получить список файлов в папке»

Действие создает переменную-список, в которой лежат пути до каждого файла в папке.

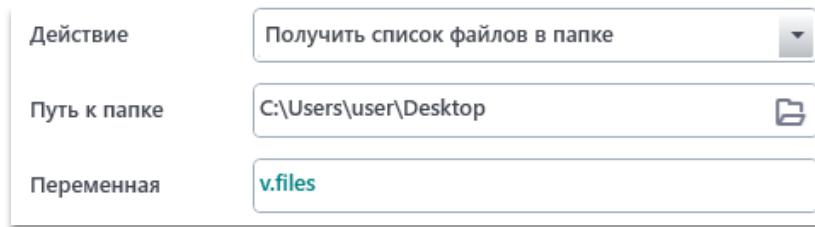


Рис. 5.13.3. Пример настройки действия «Получение списка файлов»

ПРИМЕР.

Пусть имеется папка «Робот», в которой находится 3 Excel-файла. Тогда в модуле Excel мы можем работать с каждым из файлов, например, чтобы работать с первым, нужно записывать в поле «Имя файла» `v.files[0]`.

Переменная после выполнения действия будет выглядеть как на рисунке 4.11.4:

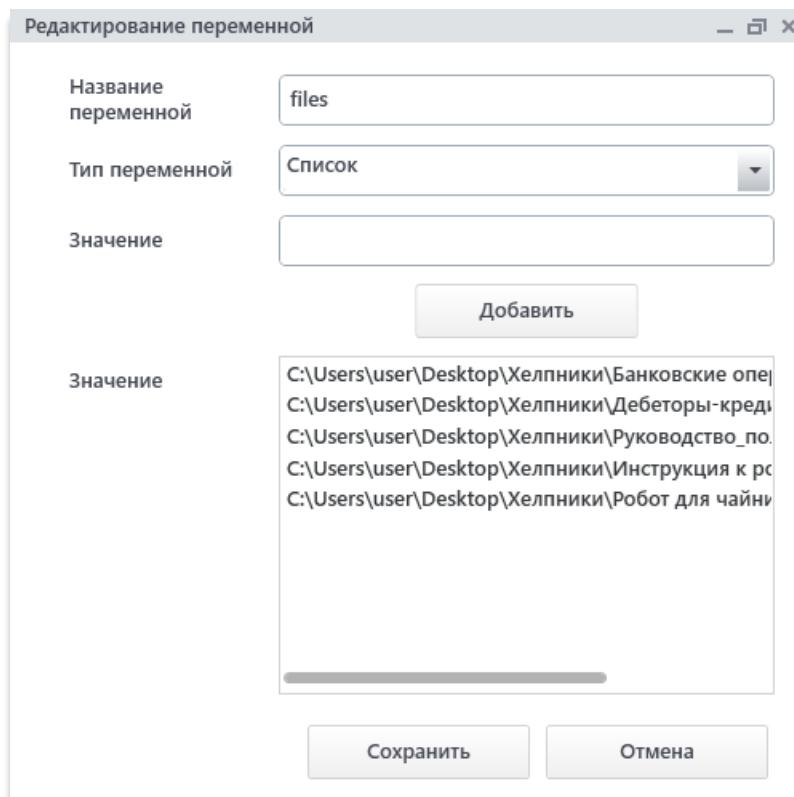


Рис. 5.13.4. Пример переменной, в которой содержится список файлов

Действие «Добавить текст в файл»

Действие добавляет текст к уже существующему в файле, не перетирая его данных.

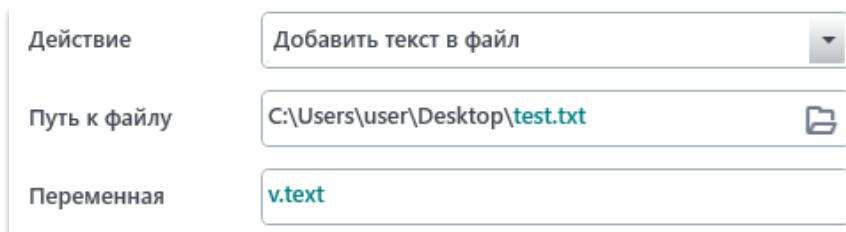


Рис. 5.13.5. Пример настройки действия «Добавить текст в файл»

Действие «Добавить переход на новую строку»

Действие добавляет символ перехода на новую строку в текст файла.

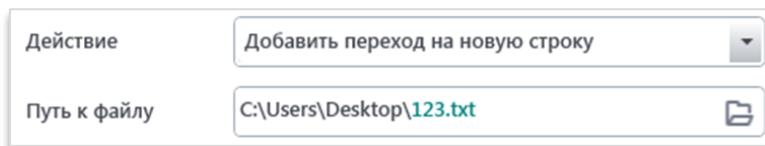


Рис. 5.13.6. Пример настройки действия «Добавить переход на новую строку»

Действие «Переместить файл»

Действие перемещает выбранный файл в указанную папку. В поле «Переменная» можно, но не обязательно, ввести название переменной, в которую будет возвращен новое расположение до файла (в ниже представленном примере – C:\Users\user\Desktop\Документы\text.docx).

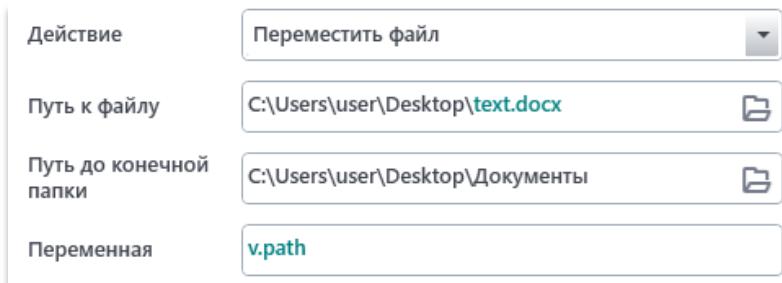


Рис. 5.13.7. Пример настройки действия «Перемещение файла»

Действие «Переименовать файл»

Действие меняет имя выбранного файла на указанное в поле «Имя файла».

Действие	Переименовать файл
Путь к файлу	C:\Users\user\Desktop\Документы\text.docx
Имя файла	newFile
Переменная	v.fileName

Рис. 5.13.8. Пример настройки действия «Переименовать файл»

Действие «Создать файл txt»

Действие создает новый файл с расширением .txt. Если поле «Переменная» будет заполнено, то в указанную в нем переменную будет возвращено расположение созданного файла.

Действие	Создать файл txt
Путь к папке	C:\Users\user\Desktop\
Имя файла	Тест
Переменная	v.path

Рис. 5.13.9. Пример настройки действия «Создать файл txt»

Действие «Удалить файл»

Действие удаляет выбранный файл.

Действие	Удалить файл
Путь к файлу	C:\Users\Desktop\Тест.txt

Рис. 5.13.10. Пример настройки действия «Удалить файл»

Действие «Скопировать файл»

Действие копирует выбранный файл в указанную директорию.

В поле «Путь до конечного файла» необходимо ввести путь до папки, в которую файл должен скопироваться, и его название с расширением так, как будто файл уже есть. Новое название может отличаться от исходного.

Если поле «Переменная» будет заполнено, то в указанную в нем переменную будет возвращено расположение нового файла.

Действие	Скопировать файл
Путь к файлу	C:\Users\user\Desktop\Тест.txt
Путь до конечного файла	C:\Users\user\Desktop\НовыйТест.txt
Переменная	v.copyFile

Рис. 5.13.11. Пример настройки действия «Скопировать файл»

Действие «Проверить существование файла»

Действие проверяет существование указанного файла и записывает результат в переменную.

«true» - если существует, и иначе «false».

Действие	Проверить существование файла
Путь к файлу	C:\Users\user\Desktop\Тест.txt
Переменная	v.checkFileExist

Рис. 5.13.12. Пример настройки действия «Проверить существование файла»

Действие «Переместить папку»

Действие перемещает папку, указанную поле «Путь к папке», в директорию, указанную в поле «Путь до конечной папки». Если поле «Переменная» будет заполнено, то в указанную в нем переменную будет возвращено новое расположение перемещаемой папки.

Действие	Переместить папку
Путь к папке	C:\Users\user\Desktop\роботы
Путь до конечной папки	C:\Users\user\Desktop\Работа
Переменная	v.folderPath

Рис. 5.13.13. Пример настройки действия «Перемещение папки»

Действие «Создать папку»

Действие создает папку с указанным названием в выбранной директории. Если поле «Переменная» будет заполнено, то в указанную в нем переменную будет возвращено расположение новой папки.

Действие	Создать папку
Путь к папке	C:\Users\user\Desktop\Работа
Имя папки	Работы
Переменная	v.folderPath

Рис. 5.13.14. Пример настройки действия «Создание папки»

Действие «Удалить папку»

Действие удаляет выбранную папку.

Действие	Удалить папку
Путь к папке	C:\Users\Desktop\работа

Рис. 5.13.15. Пример настройки действия «Удалить папку»

Действие «Проверить существование папки»

Действие проверяет существование указанной папки и записывает результат в переменную.

«true» - если существует, и иначе «false».

Действие	Проверить существование папки
Путь к папке	C:\Users\user\Desktop\Работы
Переменная	v.checkFolderPath

Рис. 5.13.16. Пример настройки действия «Проверить существование папки»

Действие «Получить список папок в папке»

Действие создает переменную-список, в которой лежат пути до каждого файла в папке.

Действие	Получить список папок в папке
Путь к папке	C:\Users\user\Desktop\Документы
Переменная	v.listFolders

Рис. 5.13.17. Пример настройки действия «Проверить существование папки»

Действие «Заархивировать папку»

Действие создает ZIP-архив с указанным именем, добавляя в него содержимое выбранной папки. Если поле «Переменная» будет заполнено, то в указанную в нем переменную будет возвращено расположение созданного архива.

Действие	Заархивировать папку
Путь к папке	C:\Users\user\Desktop\Роботы
Имя архива	Архив
Переменная	v.archive

Рис. 5.13.18. Пример настройки действия «Заархивировать папку»

Действие «Распаковать архив»

Действие распаковывает ZIP-архив в указанную папку.

Действие	Распаковать архив
Путь к файлу	C:\Users\Desktop\Robot.zip
Путь до конечной папки	C:\Users\Desktop

Рис. 5.13.19. Пример настройки действия «Распаковать архив»

ЧАСТЬ VI. МОДУЛИ-ОПЕРАТОРЫ: «ЦИКЛ», «УСЛОВИЕ», «WHILE», «TRY-CATCH»

Глава 14. Модуль «Условие»

Описываемые в этой части модули сами по себе ничего не делают, их следует использовать вместе с любыми другими. У описываемых модулей (кроме модуля «Break»), помимо привычного «Примечания», есть еще несколько параметров, обязательных для заполнения, но нет отдельного окна с настройками.

Модуль «Условие» позволяет в зависимости от указанного условия выполнять те или иные действия. Условием может быть любое выражение, которое приводится к значению «True» или «False».

Интерфейс модуля

Модуль состоит из полей «Примечание» и «Условие». Действия, которые должны выполняться при значении условия «True», добавляются в раздел «THEN», при значении «False» - в раздел «ELSE». Действия добавляются в модуль перетаскиванием их из меню или другого места рабочей области.

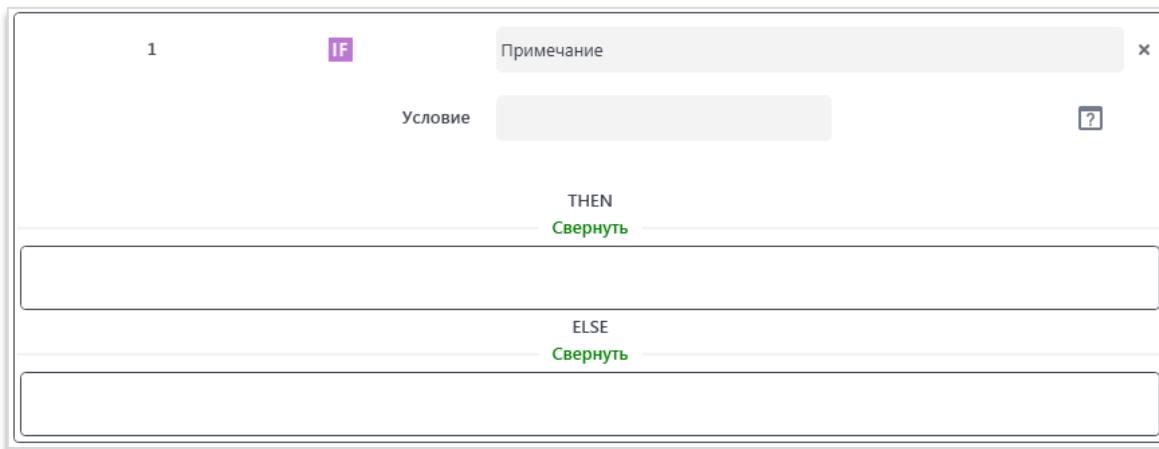


Рис. 6.14.1. Условный модуль

Поле «Примечание» предназначено для внесения комментария от разработчика робота.

Поле «Условие» заполняется по правилам языка программирования C#. Для пользователей, незнакомых с этим языком, предлагается помочь в виде «Конструктора условий», доступная по кнопке справа от условия - .

Конструктор условий

Конструктор условий позволяет составлять условия в соответствии с синтаксисом языка C#.

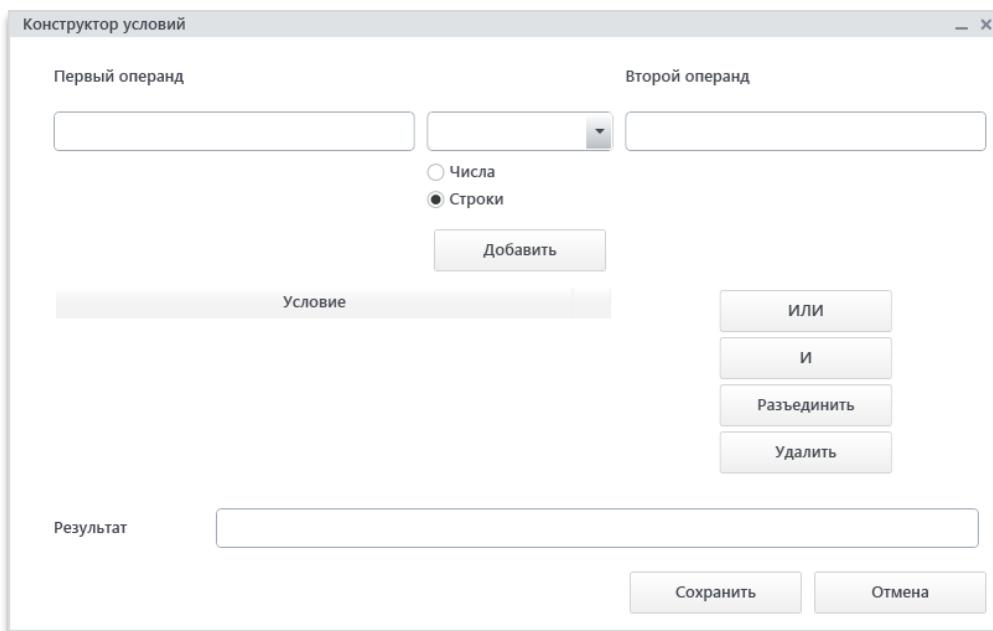


Рис. 6.14.2. Конструктор условий

Добавление

Для того чтобы составить условие, необходимо заполнить оба операнда, выбрать тип operandов (строки или числа), выбрать оператор между ними из предложенного и нажать на кнопку «Добавить»:

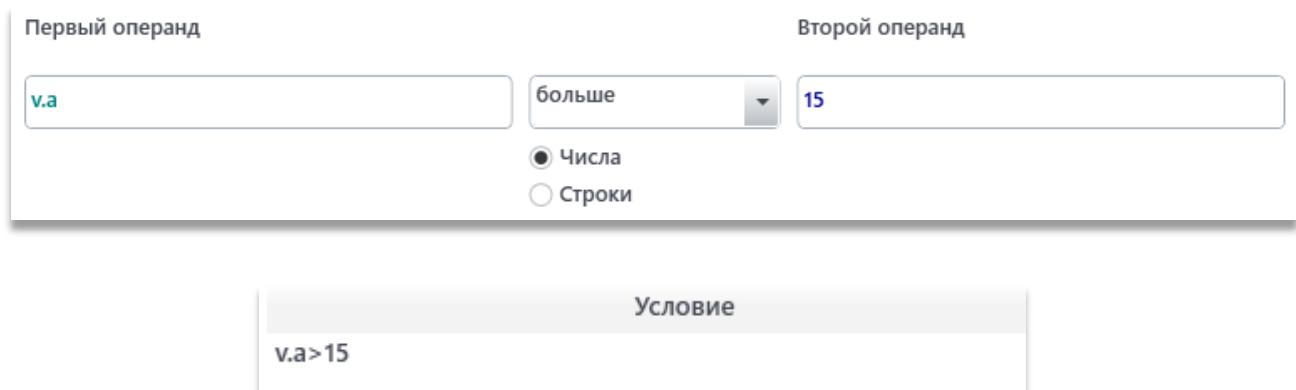


Рис. 6.14.3. Добавление условий

Список операций для строк и для чисел отличается, поэтому сначала выберите тип operandов, а лишь затем операцию.

Связывание

Несколько условий можно связать с помощью логического «И» либо логического «ИЛИ».

Делается это следующим образом – удерживая клавишу Shift, выбираются несколько условий, которые должны быть соединены одним оператором, затем нажимается кнопка с соответствующим логическим оператором:

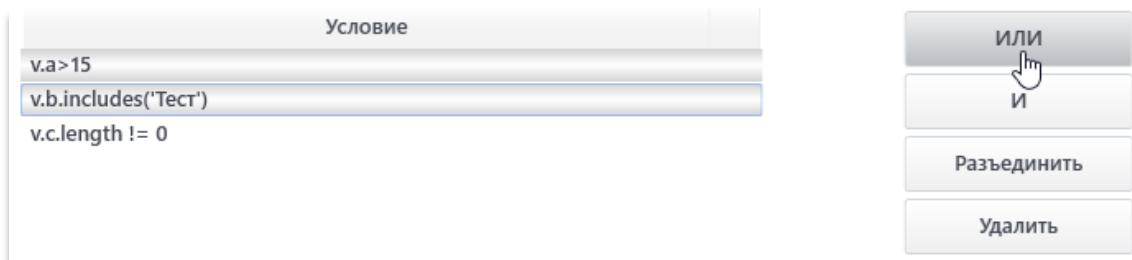


Рис. 6.14.4. Связывание условий

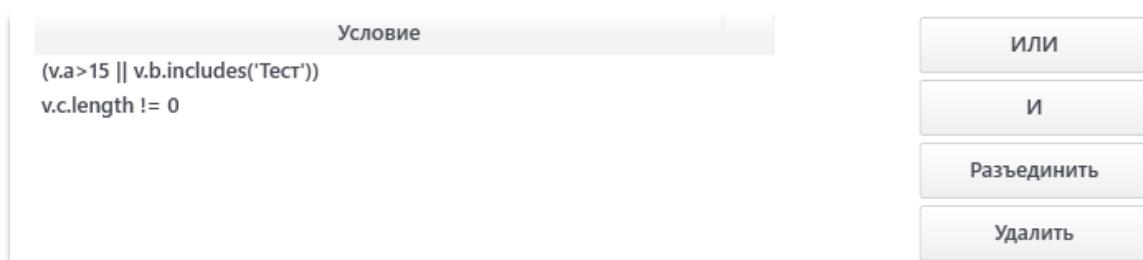


Рис. 6.14.5. Результат связывания условий

Символ «||» означает логическое ИЛИ, символ «&&» - логическое И.

Разъединение

Условия, состоящие из нескольких простых условий, то есть те, которые используют логическое И или ИЛИ, можно разъединить, нажав на соответствующую кнопку. Разъединение происходит по логическим операторам.



Рис. 6.14.6. Разъединение условий

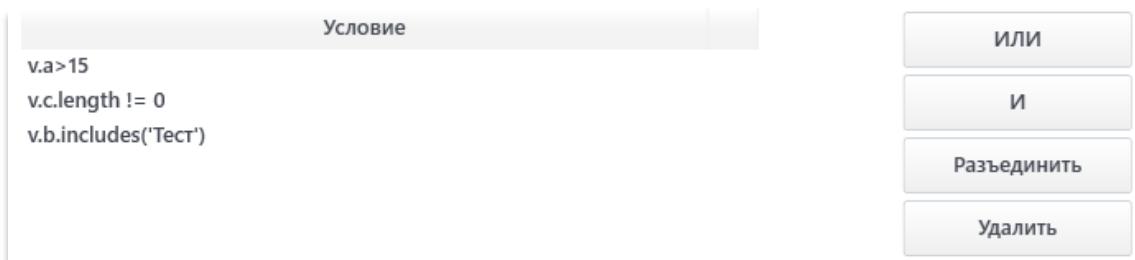


Рис. 6.14.7. Результат разъединения условий

Сохранение

Для сохранения условия необходимо нажать на кнопку «Сохранить». Получившиеся условие вставится в поле «Условие» условного модуля:

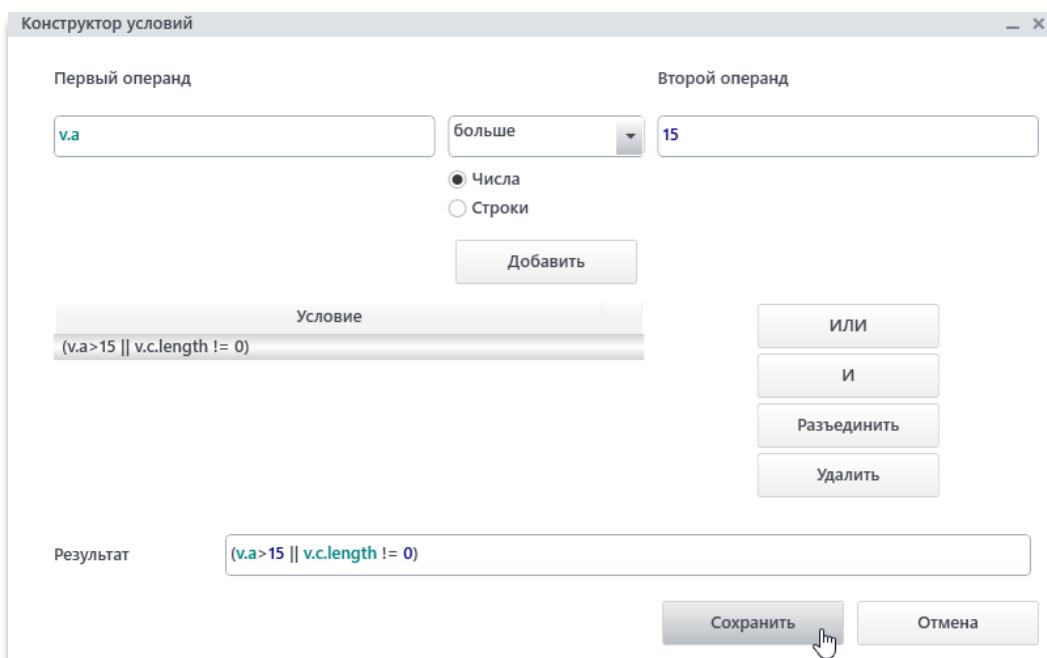


Рис. 6.14.8. Сохранение условия

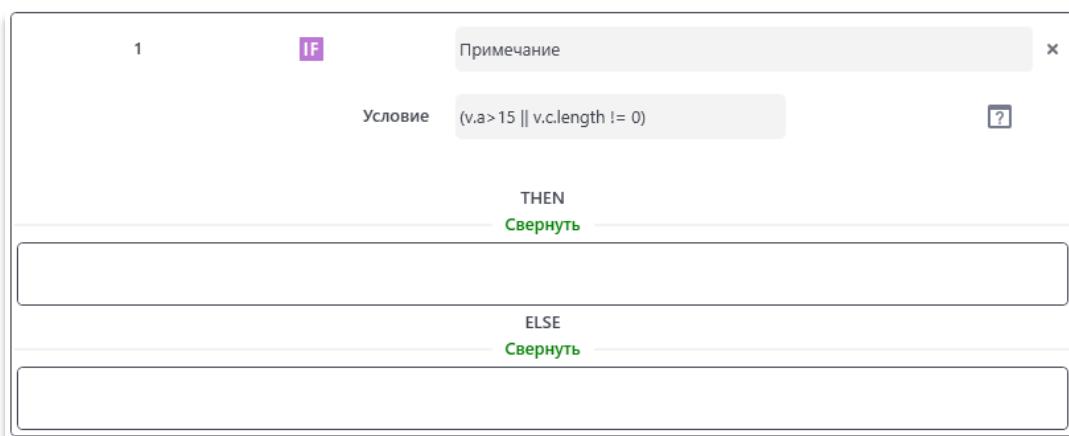


Рис. 6.14.9. Отображение условия

Законченный модуль может выглядеть следующим образом:

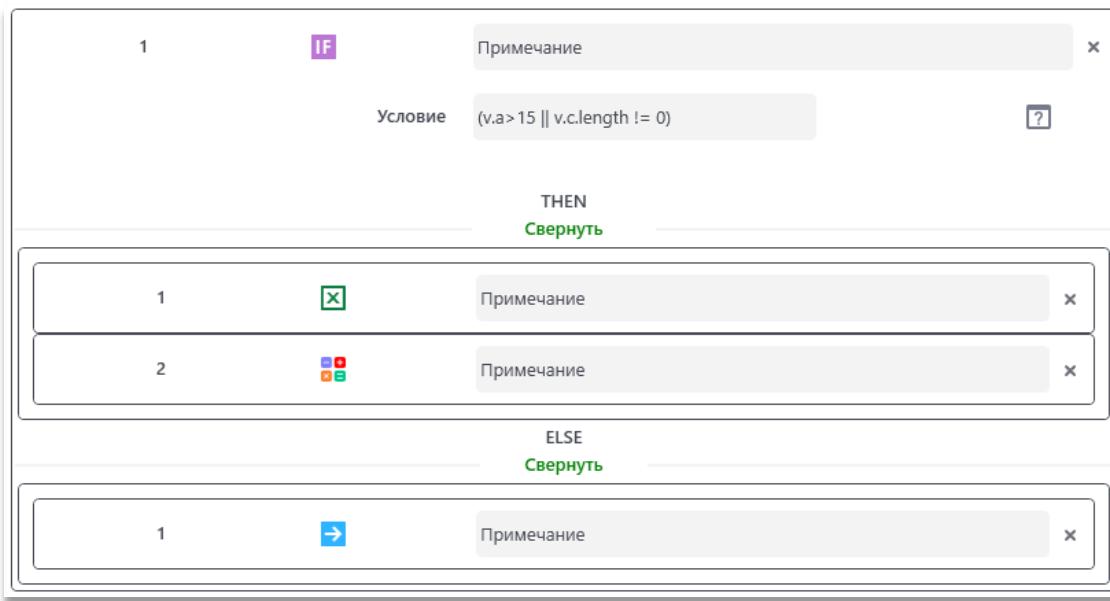


Рис. 6.14.10. Заполненный модуль «Условие»

Сворачивание

Для уменьшения занимаемого места в рабочей области проекта, можно свернуть каждый блок данного модуля по кнопке «Свернуть».

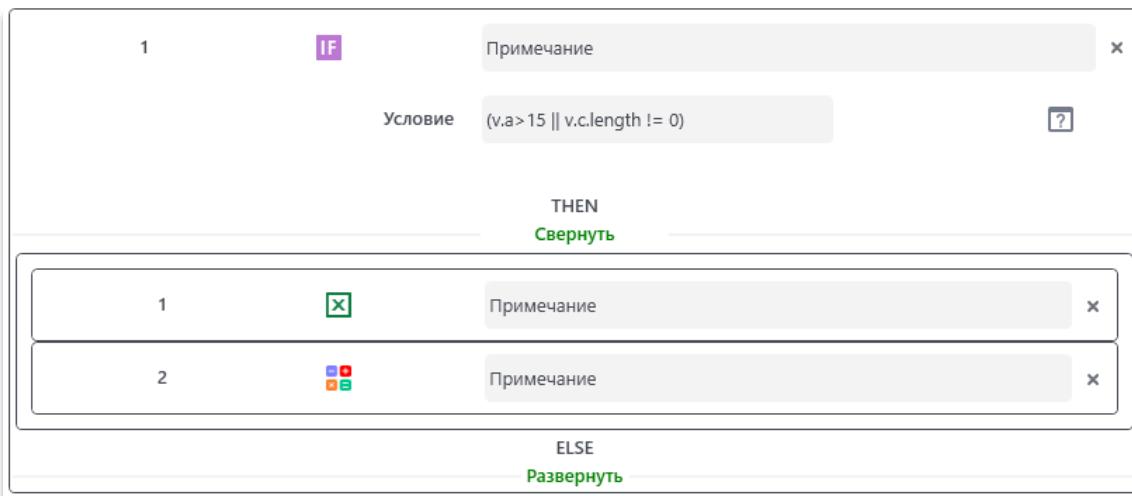


Рис. 6.14.11. Свернутый блок ELSE

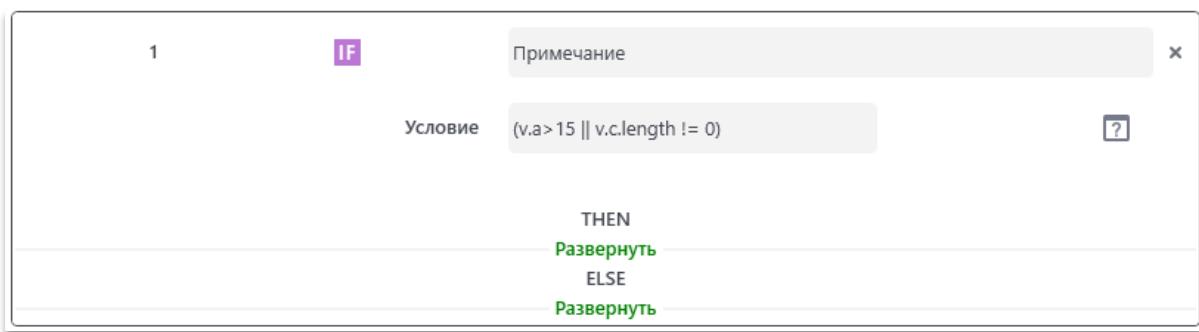


Рис. 6.14.12. Свернутые THEN и ELSE

Глава 15. Модуль «Цикл»

Модуль «Цикл» применяется для неоднократного (циклического) выполнения набора действий. Действия, добавленные в модуль, будут выполняться до тех пор, пока не пройдёт указанное в настройках количество итераций.

Интерфейс модуля

Модуль состоит из полей «Примечание», «Количество итераций» и «Переменная». Действия, которые должны выполняться в цикле, добавляются в блок, расположенный в нижней части модуля. Действия добавляются в модуль перетаскиванием их из меню или другого места рабочей области.

Модуль выглядит следующим образом:

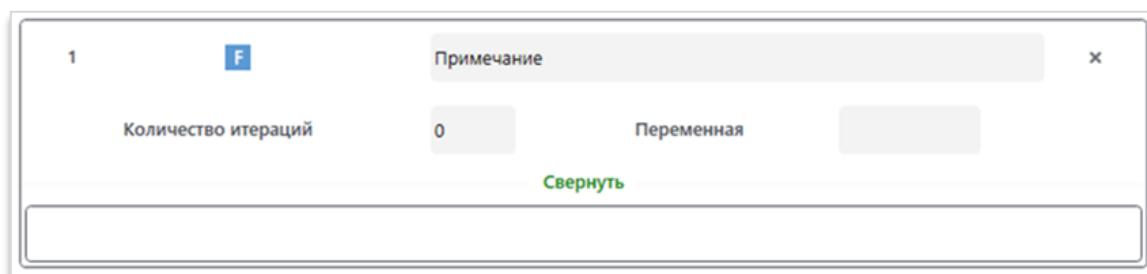


Рис. 6.15.1. Вид модуля

Поле «Примечание», как и во всех других модулях, служит для внесения комментария от разработчика робота.

Остальные поля присущи только этому модулю.

Поле «Количество итераций» предназначено для ввода числа, означающего, сколько раз необходимо выполнить действия, содержащиеся в модуле.

Поле «Переменная» - переменная, в которой будет храниться число – текущий шаг цикла. Цикл начинается с единицы.

Пример заполненного цикла представлен на Рис. 6.15.2.

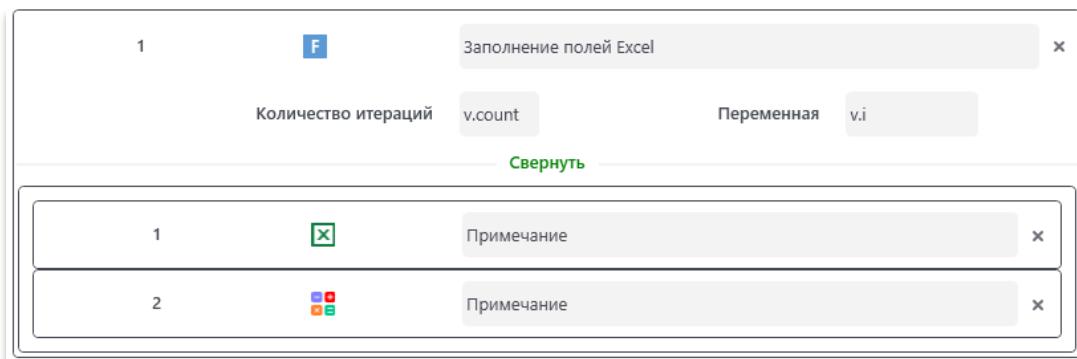


Рис. 6.15.2. Пример заполнения модуля

По кнопке «Свернуть» можно свернуть действия цикла для более компактного отображения команды.

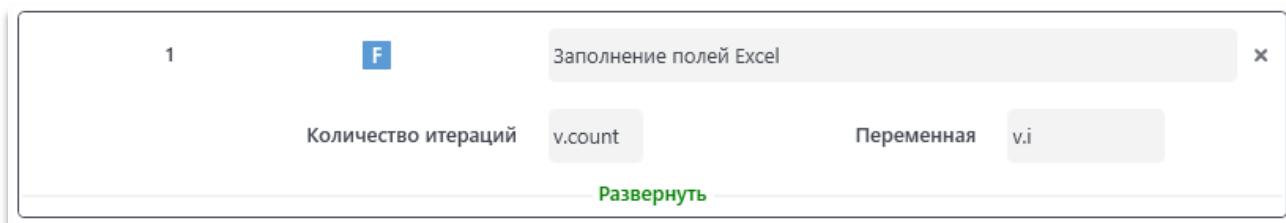


Рис. 6.15.3. Свернутый вид модуля

Глава 16. Модуль «While»

Модуль «While» (цикл с условием) выполняет добавленные в него действия до тех пор, пока указанное условие возвращает значение «True».

Интерфейс модуля

В окне модуля расположены поля «Примечание» и «Условие». В блоке, расположенном в нижней части модуля, следует располагать действия, которые требуется выполнять в цикле.

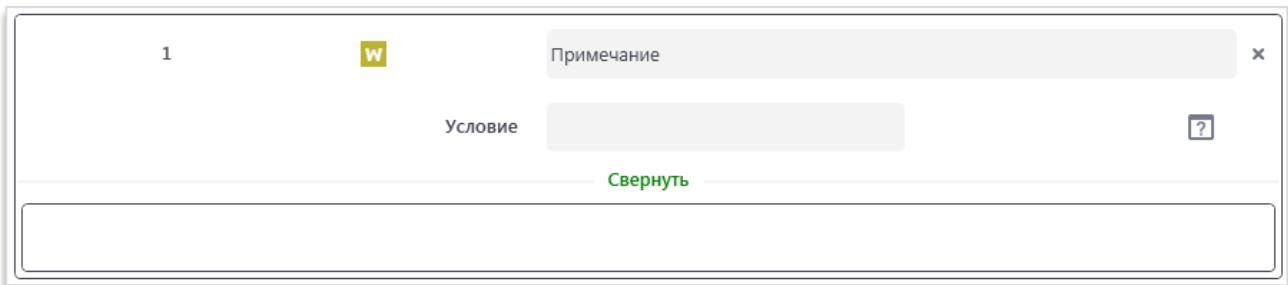


Рис. 6.16.1. Вид модуля

Поле «Условие». В поле вводится условие, как и в модуле «Условие», которое необходимо писать в соответствии с синтаксисом языка C#, поэтому здесь присутствует кнопка, открывающая «Конструктор условий» (см. Глава 14).

Добавление действий в модуль происходит с помощью перетаскивания из меню модулей или из рабочей области проекта.

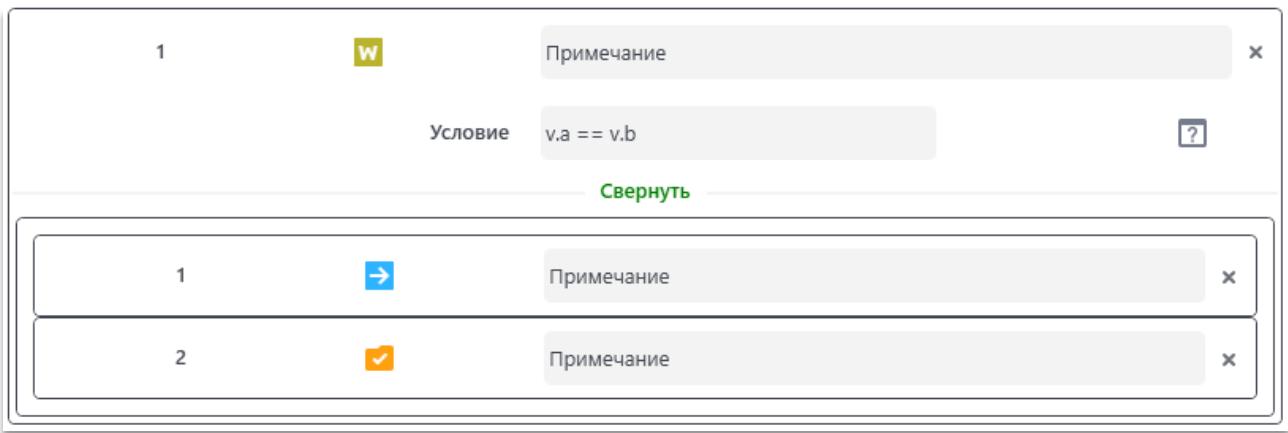


Рис. 6.16.2. Добавление действий

При нажатии на кнопку «Свернуть» действия, описанные в цикле, сворачиваются для более компактного отображения команды.

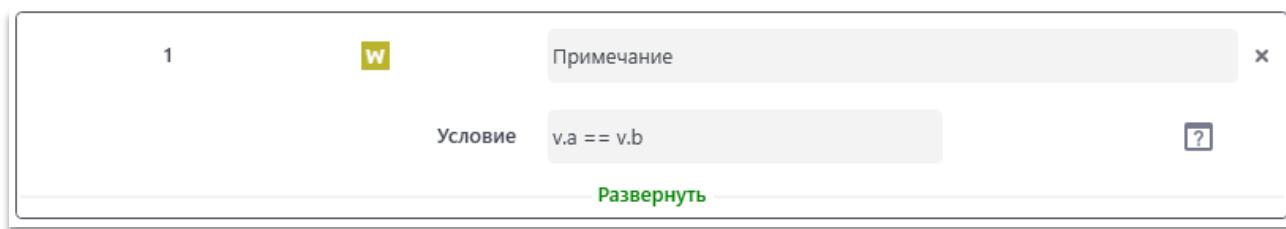
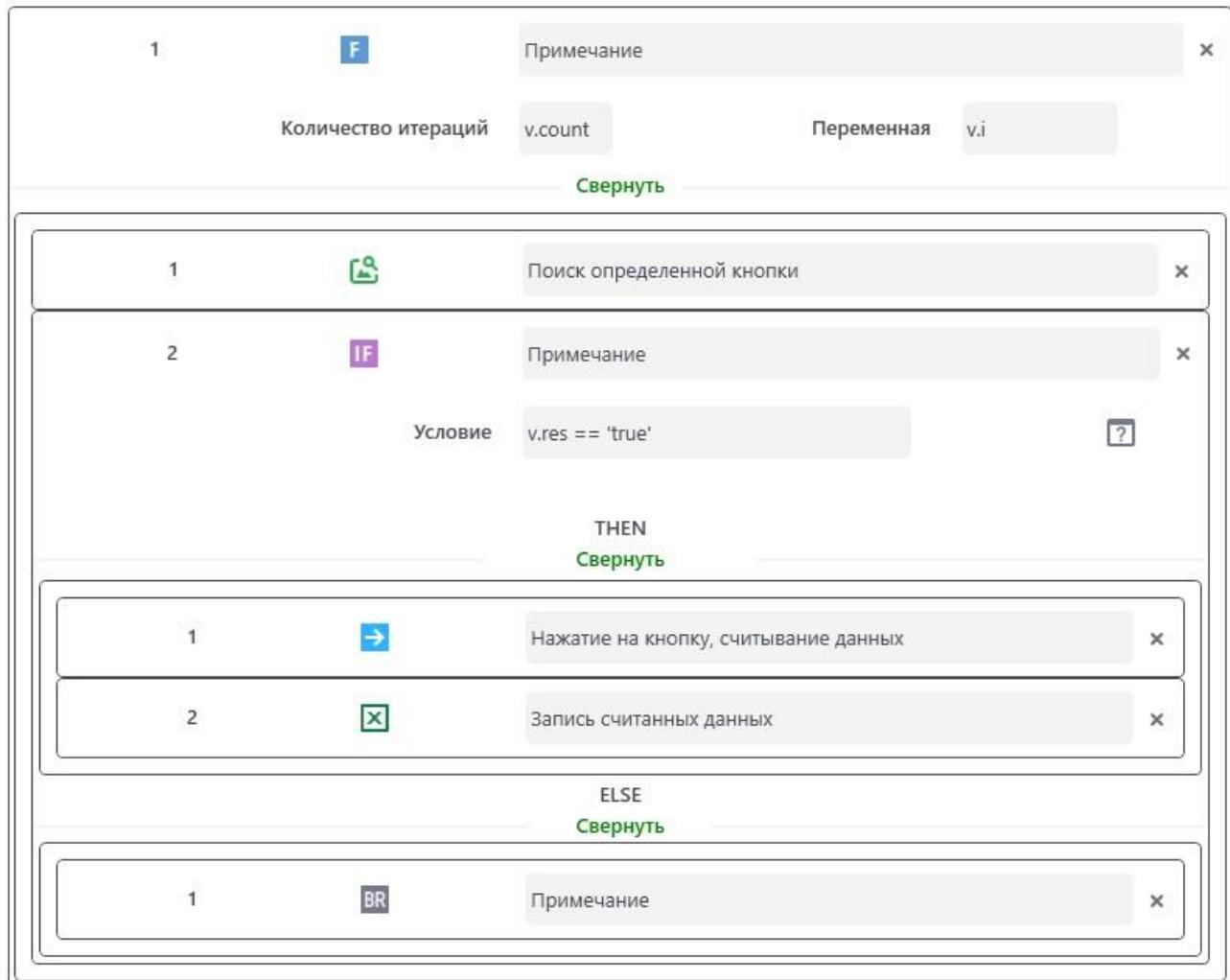


Рис. 6.16.3. Свернутый вид модуля



Глава 17. Модуль «Break»

Для «аварийного» выхода из простого цикла используют модуль «Break». Например, в цикле ведется поиск конкретного изображения, и если такое изображение найдётся, следует нажать на него, считать какие-либо данные и перейти к следующему действию, иначе – выйти из цикла и не проигрывать следующее действие. Для такого сценария необходимо создать цикл, в котором будет модуль «Условие», в одной из веток которого будет находиться модуль «Break» (модуль «Break» представлен самым последним модулем на примере):

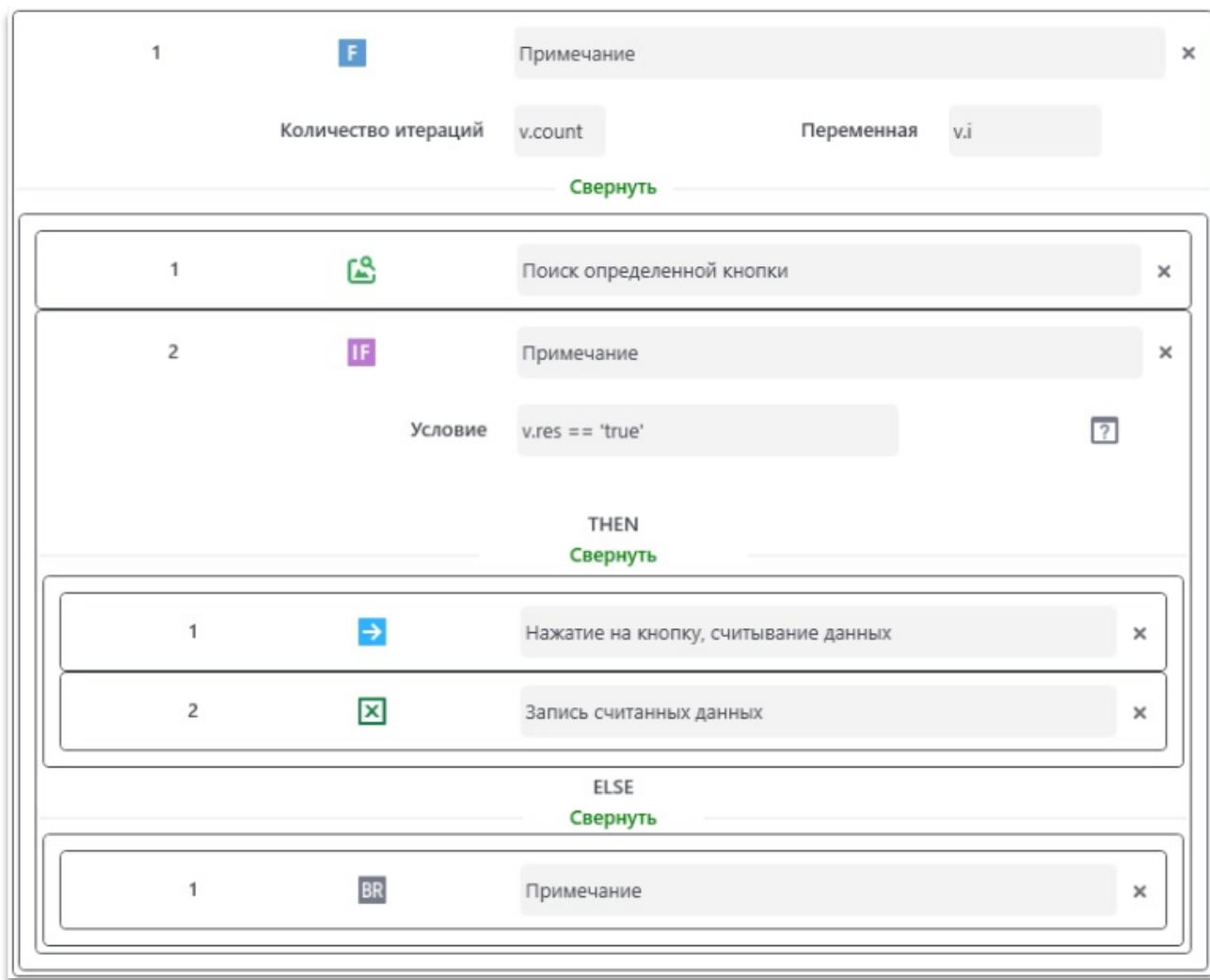


Рис. 6.17.1. Пример использования модуля «Break»

Модуль «Break» не требует настройки, но может применяться только внутри модуля «Цикл».

Глава 18. Модуль «Return»

Бывают ситуации, когда при некотором условии необходимо завершить работу робота, например, нужно обработать письмо, которое должно прийти на почту, а оно не пришло. Для таких целей существует модуль «Return».

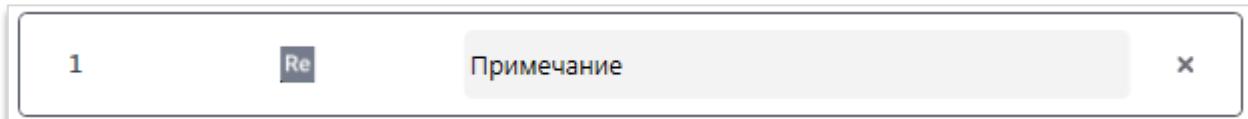


Рис. 6.18.1. Внешний вид модуля

Данный модуль заканчивает работу всего робота и не требует настройки. Может использоваться в любом месте.

ПРИМЕР.

На рисунке ниже представлен пример использования модуля: пусть с помощью модуля «Считать письма»читываются все письма за определенный промежуток времени. Все темы письма и их содержания необходимо внести в Excel-файл. Но что, если этих писем нет? Для этого проверяем наличие писем, и, если их нет, робот прекратит работу, иначе – продолжит.

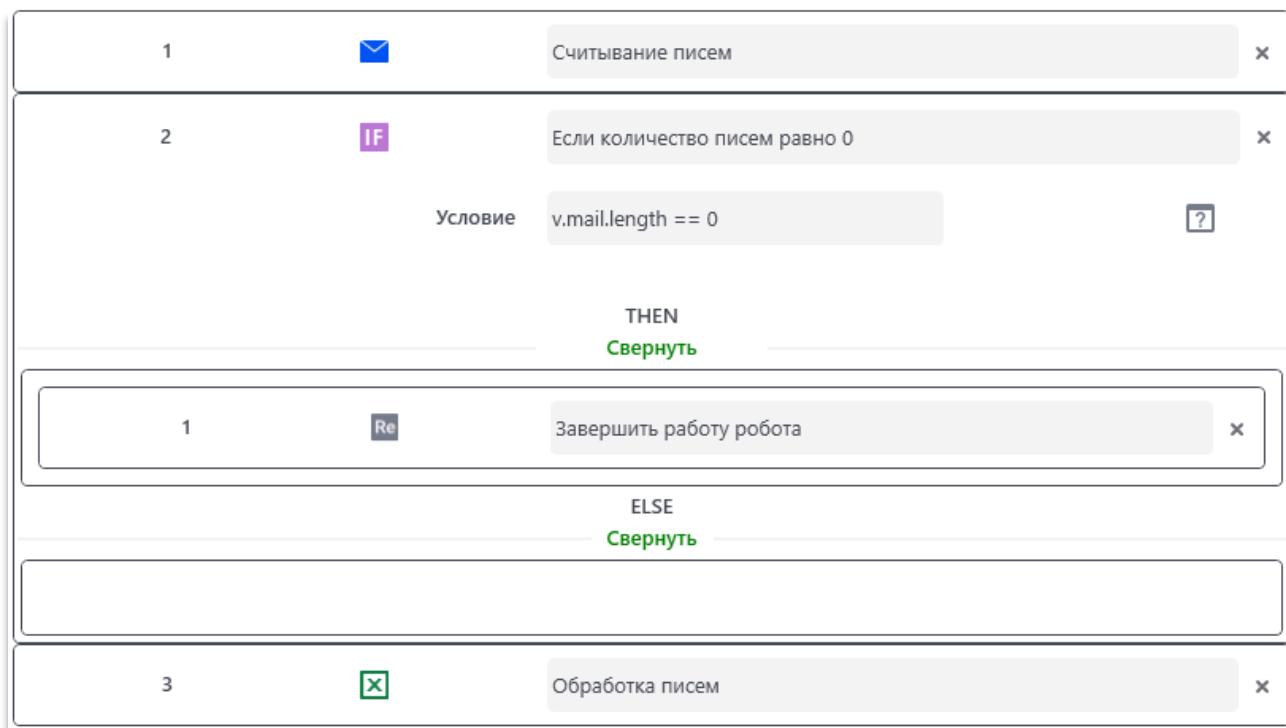


Рис. 6.18.2. Пример использования модуля «Return»

Благодаря этому модулю можно писать робота в общей рабочей области, а не в ветках модуля «Условие». Так робот выглядит более аккуратным и наглядным.

Глава 19. Модуль «Try/Catch»

Данный модуль позволяет обрабатывать ошибки, которые могут возникнуть в процессе выполнения некоторого блока модулей.

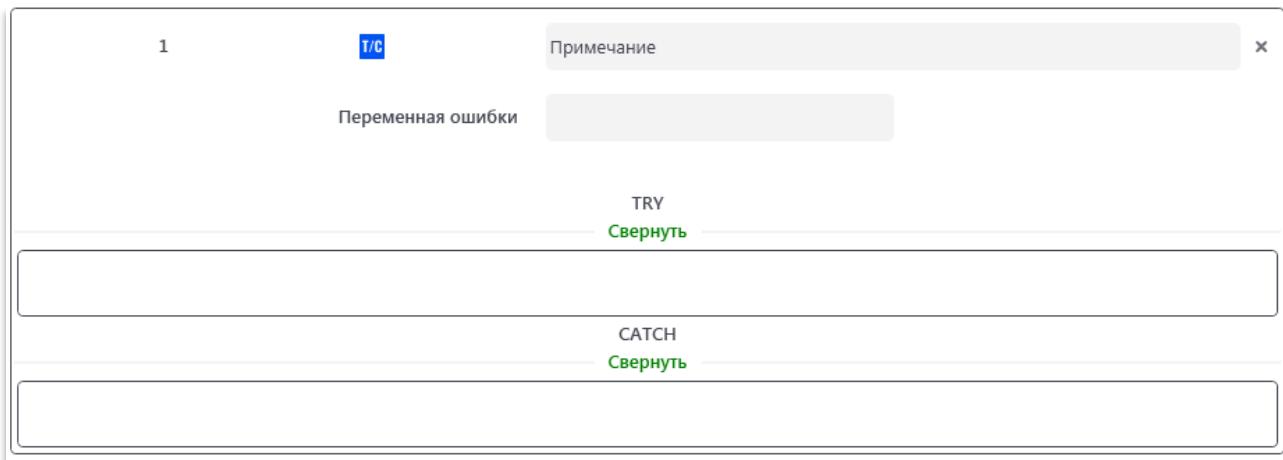


Рис. 6.19.1. Внешний вид модуля Try/Catch

Модуль состоит из двух блоков – TRY и CATCH. Блок TRY содержит набор модулей или целого робота, который должен выполниться, блок CATCH содержит набор модулей на случай, если произойдет ошибка исполнения модулей блока TRY.

Модуль имеет два текстовых поля, первое – поле «Примечание», в которое можно ввести описание модуля для более удобной навигации по роботу, а второе поле предназначено для сохранения возникшей в модуле TRY ошибки. Так как есть возможность узнать, какая ошибка возникла, каждую возможную ошибку можно корректно обработать в блоке CATCH.

ПРИМЕР.

Пусть стоит задача – скачать письмо с почты с определенным названием вложения в заранее заданную папку и обработать это вложение модулем Excel, причем в модуле Excel путь до файла будет всегда одинаковым. Например, вложение называется «Отчет за день», путь до папки «C:\Reports», тогда в Excel будет всегда открываться файл «C:\Reports\Отчет за день.xlsx», а вложения с почты всегда будут скачиваться в вышеназванную папку.

Если письмо с необходимым вложением еще не пришло, то модуль Excel выдаст ошибку «Не удалось открыть документ C:\Reports\Отчет за день.xlsx». В таком случае необходимо как-то обработать эту ошибку, предложить пользователю запустить робота позже, или сказать ему подождать в течение некоторого времени и попробовать снова.

Для обработки ошибки с выводом информационного окна необходимо добавить модуль «Try/Catch», где в блоке TRY будут содержаться модули считывания письма и Excel, а в блоке CATCH модуль «Интерфейс» для вывода пользовательского окна.

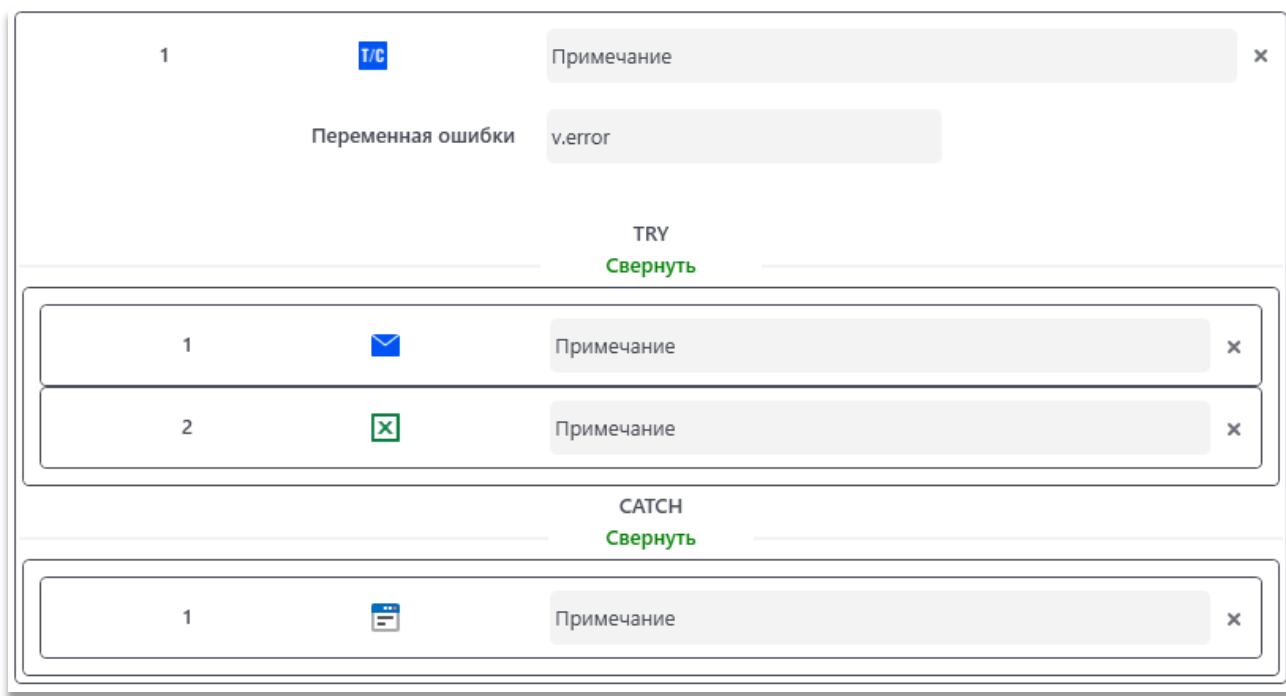


Рис. 6.19.2. Пример использования модуля «Try/Catch»

В результате работы такого робота будет либо стандартное сообщение «Скрипт отработал успешно!», если письмо было на почте, либо окно с текстом «Письмо еще не пришло» в ином случае. В переменной v.error будет содержаться текст «Не удалось открыть документ C:\Reports\Отчет за день.xlsx».

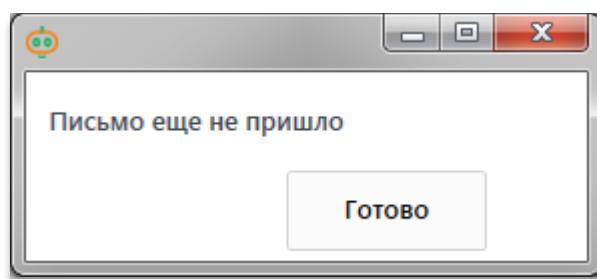


Рис. 6.19.3. Пример обработки ошибки – окно с информацией

ЧАСТЬ VII. МОДУЛИ «ПОЧТОВЫЕ АГЕНТЫ»

Глава 20. Модуль «Отправить письмо»

Модуль «Отправить письмо» предназначен для отправки письма одному или нескольким получателям.

Интерфейс модуля

Окно модуля состоит из нескольких вкладок: «Настройки», «Получатель», «Письмо». Рассмотрим по порядку все вкладки. Начнем знакомство с рассмотрения вкладки «Настройки».

Вкладка «Настройки»

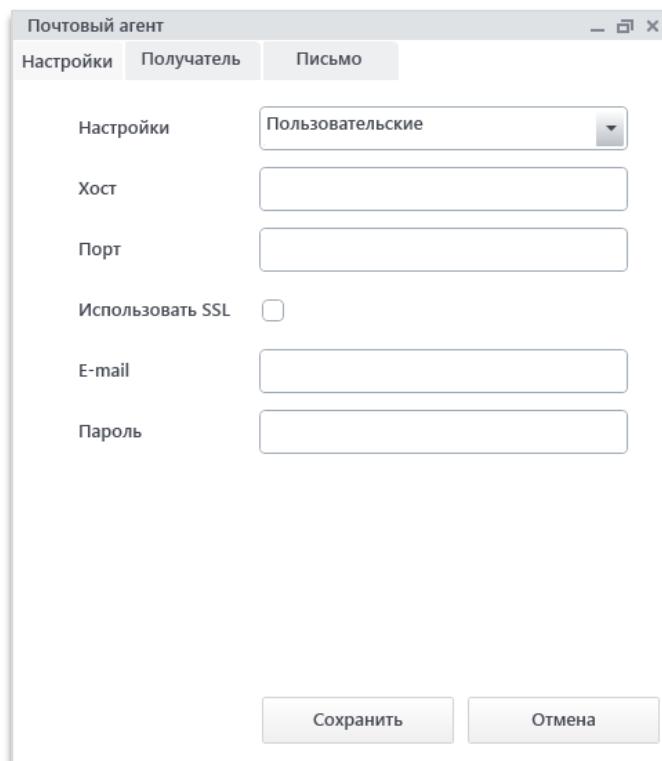


Рис. 7.20.1. Окно модуля

Поле «Настройки» – выпадающий список, в котором предложены несколько почтовых служб. При выборе почтовой службы будут заполнены поля «Хост», «Порт» и установлен флагок «Использовать SSL» согласно её настройкам. Пункт "Пользовательские" предоставляет возможность самостоятельной настройки параметров доступа к почтовой службе.

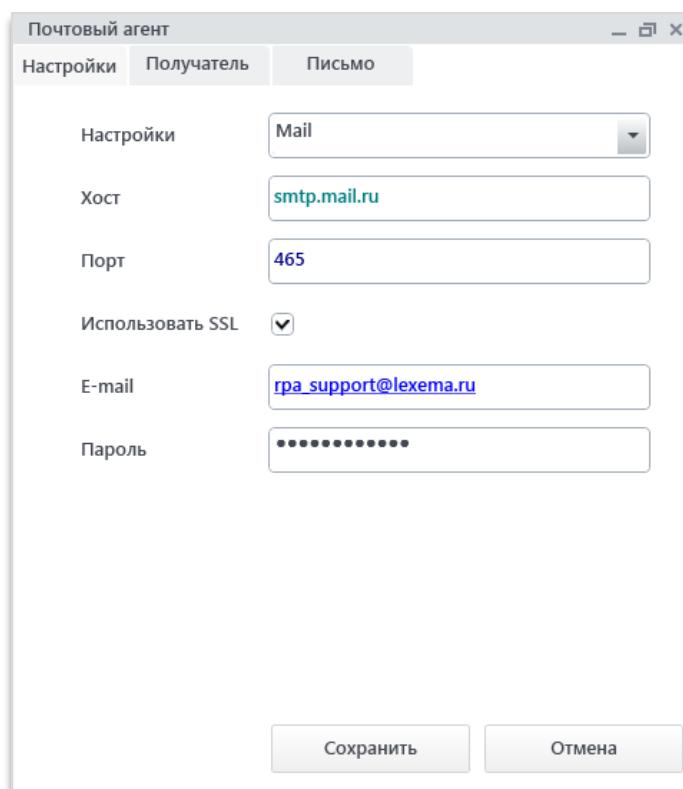
Настройки	Mail
Хост	smtp.mail.ru
Порт	465
Использовать SSL	<input checked="" type="checkbox"/>

Рис. 7.20.2. Пример настройки почтовой службы «Mail»

Поля «Хост», «Порт» и флагок «Использовать SSL» необходимы к заполнению только при выборе пользовательских настроек. Представляют собой набор обязательных полей для подключения к почтовым сервисам.

Поле «E-mail» предназначено для указания адреса электронной почты, с которой будут отправлены сообщения, в формате name@mail.ru.

Поле «Пароль» предназначено для ввода пароля от почты, указанной в поле «E-mail».



Настройки	Mail
Хост	smtp.mail.ru
Порт	465
Использовать SSL	<input checked="" type="checkbox"/>
E-mail	rpa_support@lexema.ru
Пароль	*****

Рис. 7.20.3. Пример заполненной вкладки «Настройки»

Вкладка «Получатель»

На данной вкладке заполняется информация о лицах, которым будет отправлено письмо.

Модуль предоставляет различные способы указания получателей, а именно:

- ручной ввод списка получателей (имена и e-mail'ы);
- из переменной-элемента, в виде одного e-mail'a;
- из переменной-списка, в виде нескольких e-mail'ов.

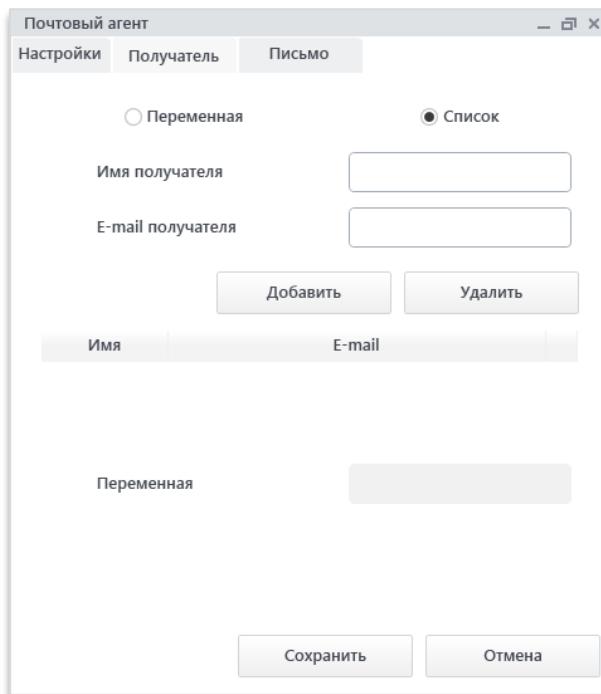


Рис. 7.20.4. Вкладка «Получатель»

Если выбран переключатель «Список», то поле «Переменная» недоступно для ввода, иначе – наоборот, доступно поле «Переменная», но недоступны другие поля.

Для ввода получателей методом «Список» необходимо заполнить поля «Имя получателя» и «E-mail получателя», затем нажать на кнопку под полями «Добавить». После этого введенные значения появятся в таблице в центре окна. Если необходимо удалить какого-либо получателя из таблицы, необходимо выбрать строку с ним и нажать на кнопку «Удалить».

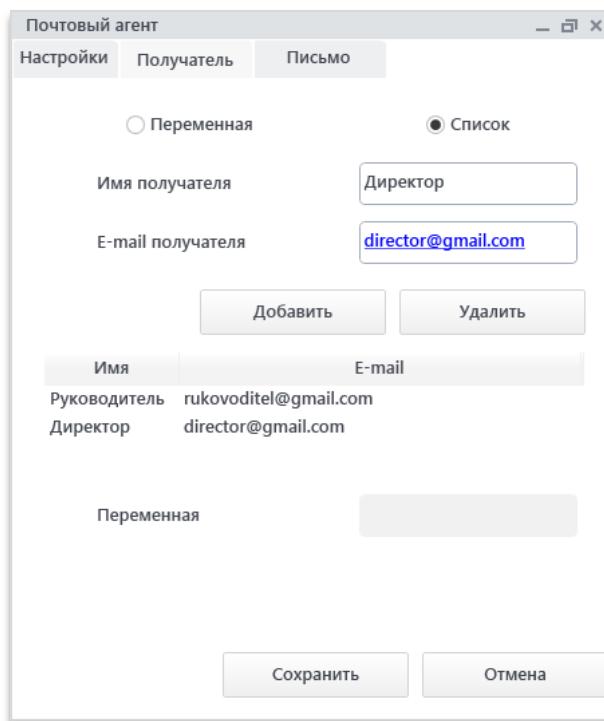


Рис. 7.20.5. Пример заполнения таблицы получателей

Если используется метод выбора получателей через переменную, то поля «Имя получателя», «E-mail получателя» и таблица становятся недоступными. В поле «Переменная» необходимо ввести название переменной, начиная с символов «v.», из которой будут взяты email'ы получателей. Переменная может быть списком из email'ов, либо строкой с перечислением их через запятую.

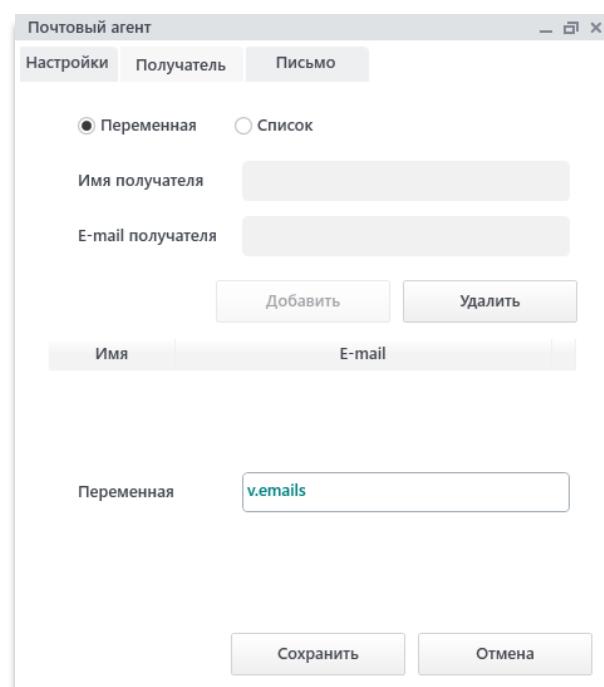


Рис. 7.20.6. Вкладка «Получатель»

Вкладка «Письмо»

На данной вкладке настраивается письмо, которое будет отправлено получателям, указанным на вкладке «Получатель». Все поля необязательны для заполнения, если не заполнить ни одного из них будет отправлено пустое письмо с темой «<Без темы>».

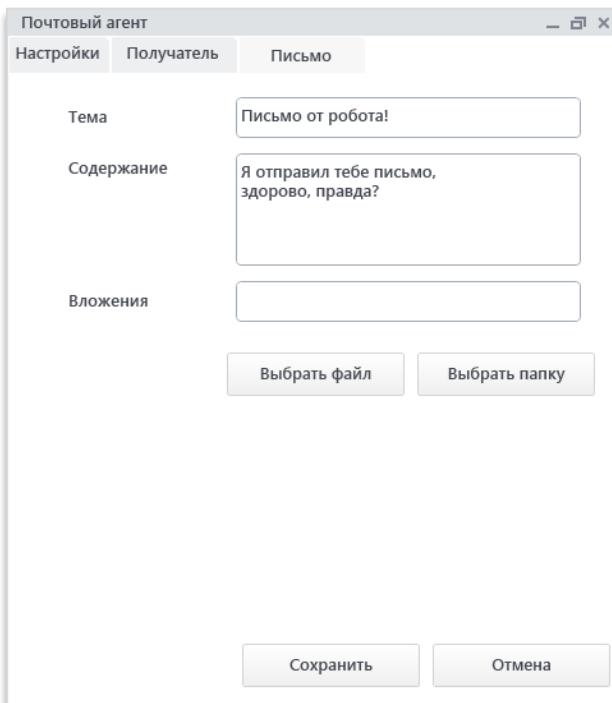


Рис. 7.20.7. Вкладка «Письмо»

Для прикрепления одного файла во вложение следует нажать на кнопку «Выбрать файл» и в открывшемся диалоговом окне выбрать требуемый файл.

Для прикрепления сразу нескольких файлов необходимо нажать на кнопку «Выбрать папку» и в открывшемся диалоговом окне выбрать папку со всеми требуемыми файлами.

Каждое из полей формы может быть заполнено переменной.

Глава 21. Модуль «Считать письма»

Модуль «Считать письма» предназначен для получения списка писем с почты для дальнейшей их обработки или просто для скачивания вложений. Для работы модуля обязательно разрешение подключения по IMAP или POP3 протоколу на используемой почте (подробнее про включение протоколов описано этой главе, пункте «Включение IMAP и POP3 протоколов»).

Интерфейс модуля

Модуль состоит из двух вкладок: «Настройки» и «Настройки чтения».

Вкладка «Настройки» аналогична одноименной вкладке в модуле «Отправить письмо», с одним лишь отличием - в данном модуле предоставляется выбор протокола отправки: IMAP или POP3. При выборе соответствующего протокола и его настроек, поля «Хост», «Порт» и «Использовать SSL» заполняются автоматически (кроме режима «Пользовательские настройки»).

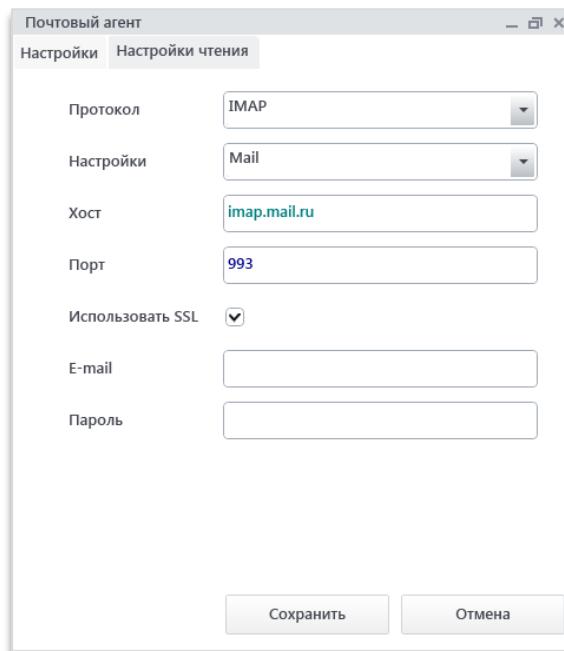


Рис. 7.21.1. Окно модуля с примером авто-заполнения полей

Вкладка «Настройки чтения»

Вкладка «Настройки чтения» представляет собой набор полей, которые заполняются в зависимости от того, какие письма следует считать.

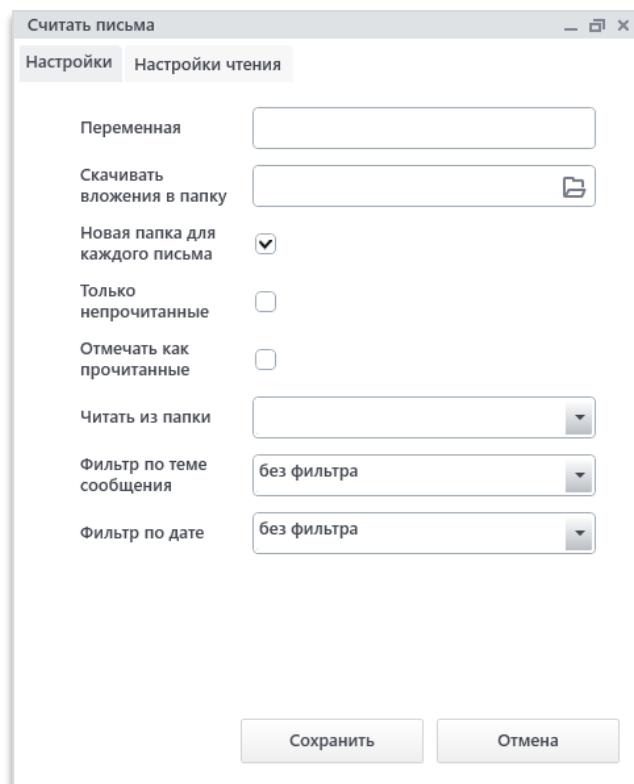


Рис. 7.21.2. Вкладка «Настройка чтения»

Поле «Переменная». В поле ожидается ввод названия переменной, в которую будет внесена информация о считанных письмах. В программе это поле будет представлено списком объектов, имеющих шесть полей:

- 1) Subject – тема письма;
- 2) Body – тело письма, его основной текст;
- 3) Sender – отправитель;
- 4) Attachments – список путей до скачанных вложений;
- 5) Date – дата получения письма;
- 6) Id – уникальный идентификатор письма.

Примеры работы с переменными

- v.mails[2].Attachments[0] – получение пути первого вложения третьего считанного письма:

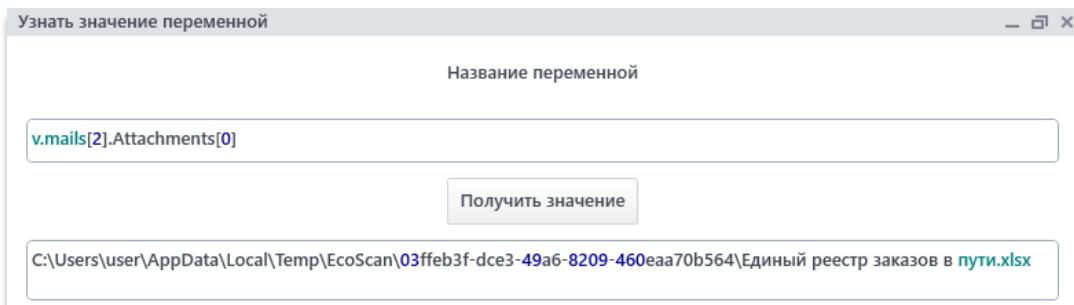


Рис. 7.21.3. Пример получения пути до первого вложения второго письма

- `v.mails[3].Subject` – получение темы четвертого скачанного сообщения:

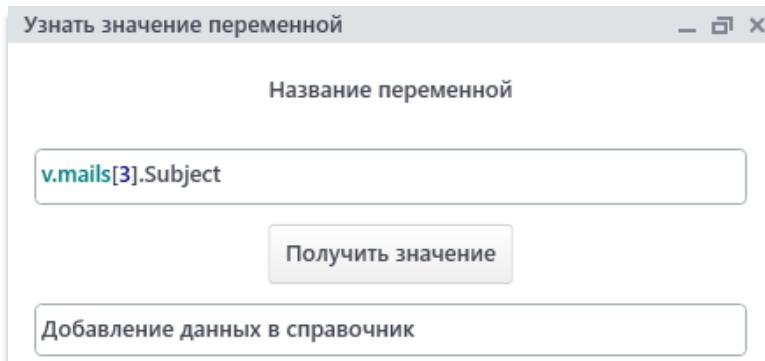


Рис. 7.21.4. Пример получения темы первого письма

Поле «Скачивать вложения в папку» предназначено для ввода пути до папки, в которую загружаются все вложения считанных писем. Если оставить поле пустым, то файлы будут загружаться во временную директорию компьютера.

Поле «Новая папка для каждого письма» позволяет сохранять вложения из каждого письма в свою отдельную папку для того, чтобы файлы, имеющие одинаковые названия, но содержащиеся в разных письмах, не перетирали друг друга. При скачивании вложений во временную системную папку компьютера галочка является обязательной.

Флажок «Только непрочитанные» позволяет настроить считывание только новых писем. *Доступно только для протокола IMAP*.

Флажок «Отмечать как прочитанные» позволяет после считывания устанавливать пометку о прочтении письма. *Доступно только для протокола IMAP*.

Поле «Читать из папки» предназначено для настройки конкретной папки в почтовом ящике, из которой следует считывать письма. По умолчанию, письма считаются из папки «Входящие». *Доступно только для протокола IMAP*.

Поле «Фильтр по теме сообщений» предназначено для настройки фильтра по теме сообщений. Если необходимо скачать письма с темой, содержащей или не содержащей определенную строку, то следует выбрать одноименную операцию в выпадающем списке и вписать нужную строку в появившемся поле:

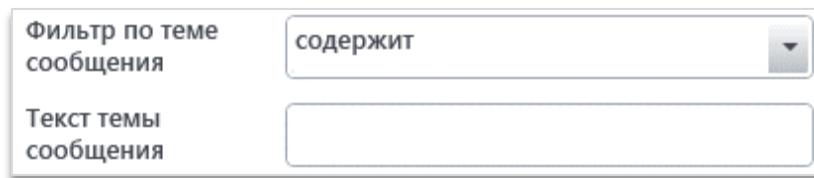


Рис. 7.21.5. Фильтр по теме сообщений

Поле «Фильтр по дате» предназначено для настройки фильтрации считывания писем по дате. При указании значения в данном поле, робот будет считывать только те сообщения, которые удовлетворяют заданному условию.

Доступны следующие методы фильтрации:

- за определенную дату – «На дату»;
- начиная с какой-либо даты, включая её – «С даты»;
- все письма, до какой-либо даты, включая её – «До даты»;
- «Между датами».

После выбора метода фильтрации под полем «Фильтр по дате» появится одно или два поля для внесения в них нужных значений в формате ДД.ММ.ГГГГ.

Включение IMAP и POP3 протоколов

1. Яндекс

1.1. IMAP

Для использования протокола IMAP в почтовой службе Яндекс необходимо перейти в настройки почты (1), затем «Почтовые программы» (2) и разрешить доступ к почтовому ящику с помощью почтовых клиентов с сервера imap.yandex.ru (3), способы авторизации – по портальному паролю (4).

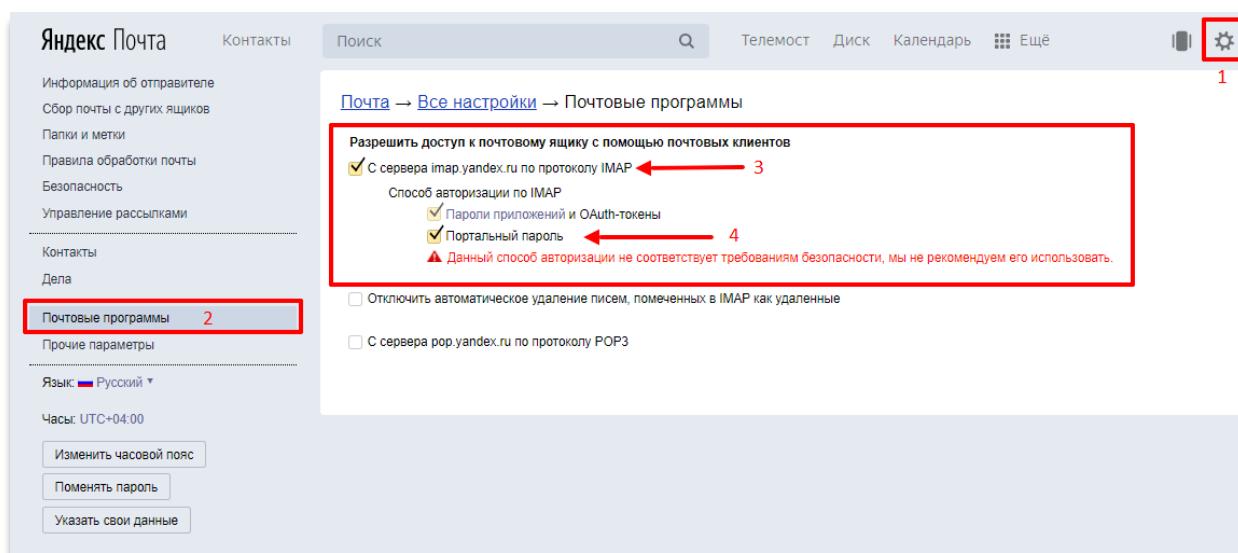


Рис. 7.21.6. Включение IMAP протокола в Яндекс

Если настройки «Портальный пароль» нет, значит в вашем аккаунте предусмотрено использование паролей приложений, которые для работы Lexema-RPA Studio необходимо выключить. Для этого перейдите в управление аккаунтом - «Пароли и авторизация» - «Пароли приложений» - «Выключить пароли приложений».

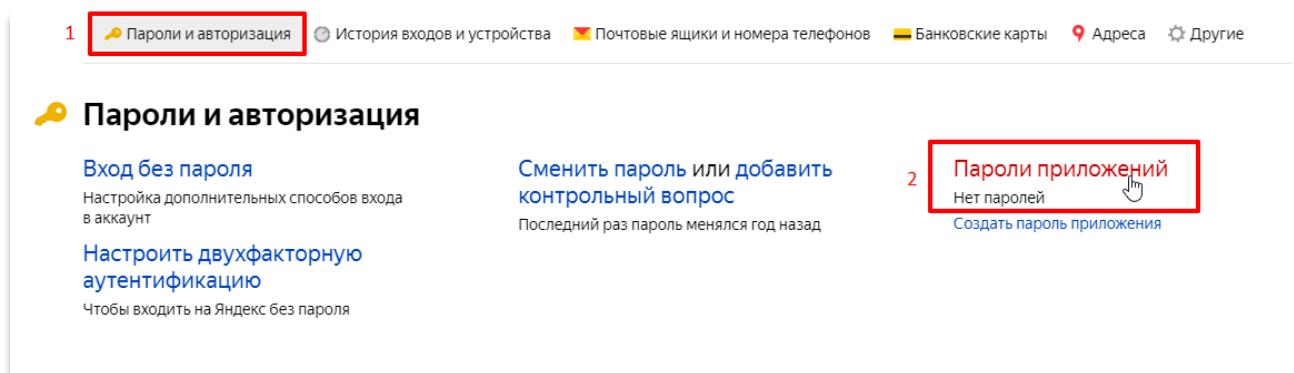


Рис. 7.21.7. Пароли приложений в настройках аккаунта

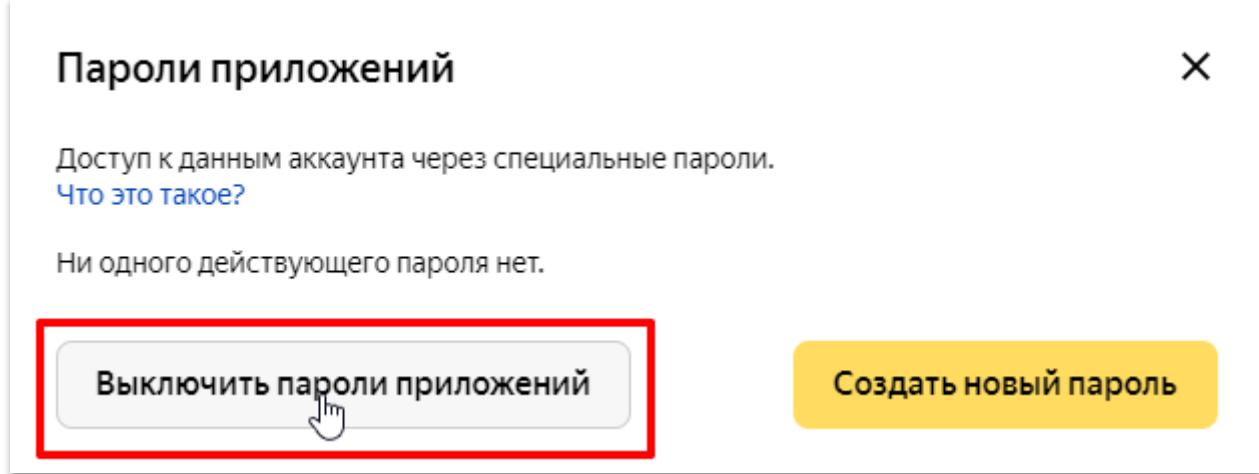


Рис. 7.21.8 Выключение паролей приложения

Затем вернитесь на страницу с включением настроек почтовых программ и установите галочку напротив «Портального пароля», как это изображено на Рис. 7.21.6. Включение IMAP протокола в Яндекс.

1.2. POP3

Для включения протокола POP3 необходимо войти в те же настройки, что и при включении протокола IMAP: «Настройки» (1) – «Почтовые программы» (2) - разрешить доступ к почтовому ящику с помощью почтовых клиентов с сервера pop.yandex.ru (3).

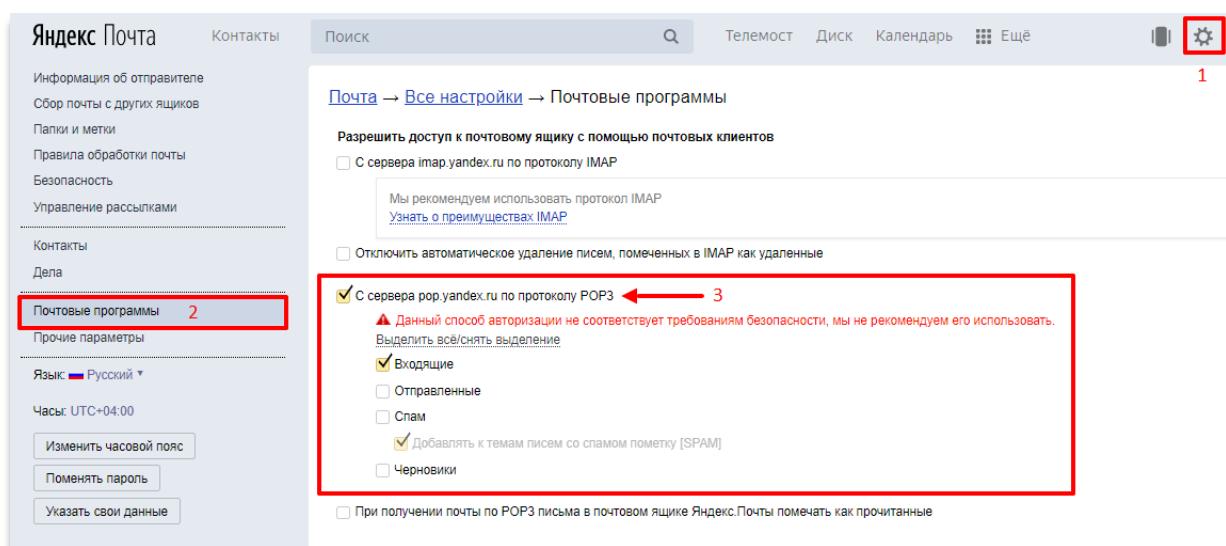


Рис. 7.21.9. Включение протокола POP3 в Яндекс

2. GMail

2.1. IMAP

Для включения IMAP протокола в почтовой службе GMail необходимо перейти в настройки почты (1), затем выбрать пункт «Пересылка и POP/IMAP» (2) - «Доступ по IMAP» и затем в «Состоянии» включить IMAP (3):

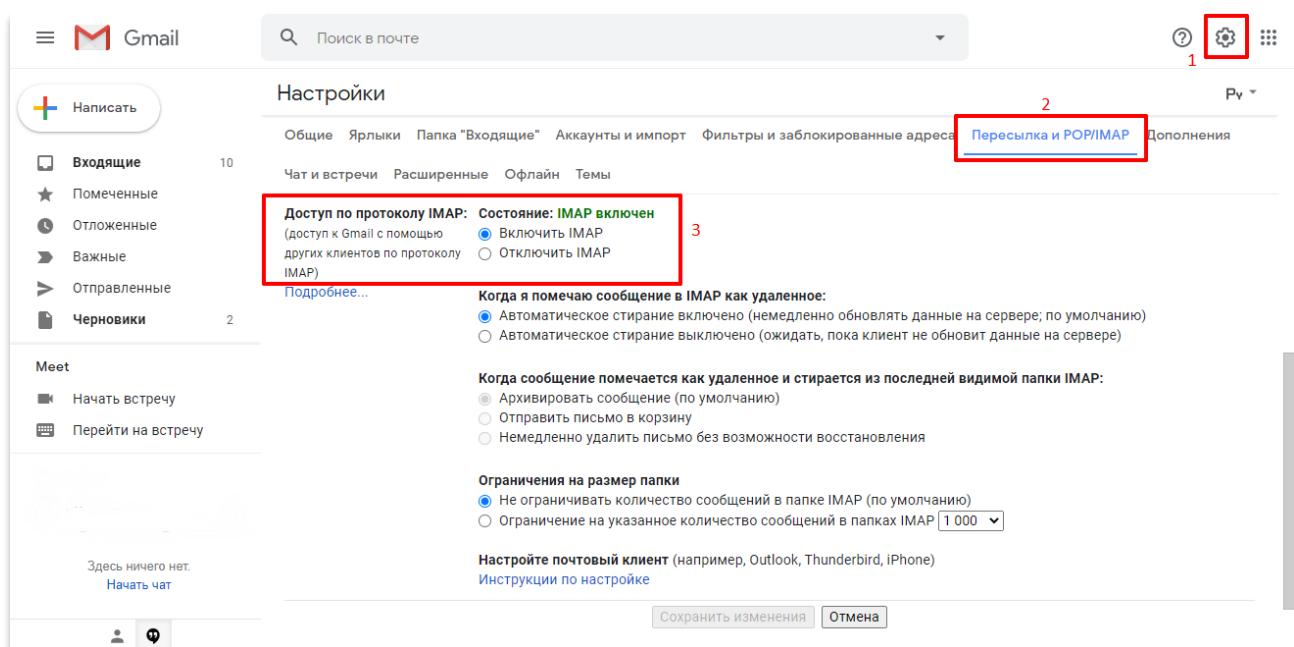


Рис. 7.21.10. Включение протокола IMAP в Gmail

2.2. POP3

Аналогично проводится включение протокола POP3: настройки (1) – «Пересылка и POP/IMAP» (2) - «Доступ по протоколу POP» - «Включить POP для всех писем» (3):

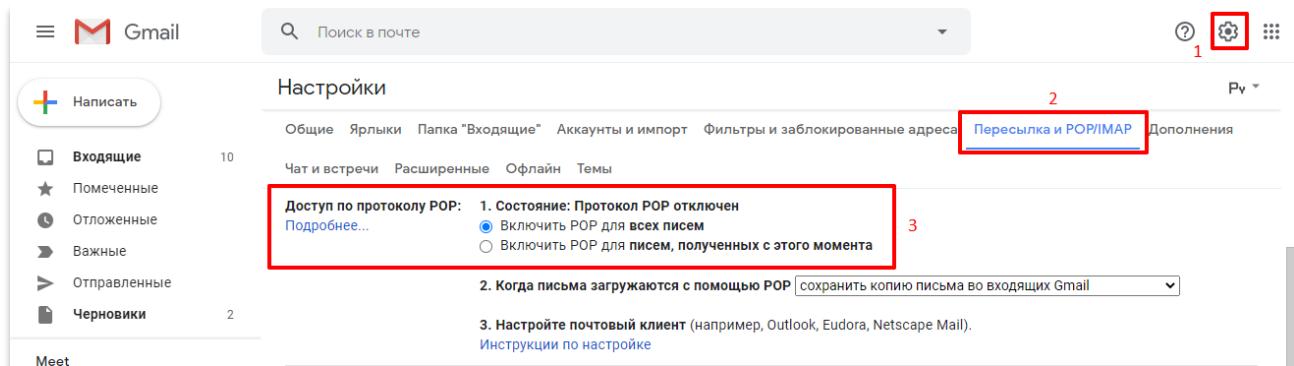


Рис. 7.21.11. Включение POP3 протокола в Gmail

Глава 22. Переместить письмо

С помощью данного модуля можно переносить уже считанные письма в другие папки в почте. Для использования модуля предварительно необходимо получить письма с помощью модуля «Считать письма».

Интерфейс модуля

Интерфейс модуля состоит из двух вкладок – «Настройки» и «Параметры перемещения».

Вкладка «Настройки»

Вкладка настроек не отличается от аналогичной вкладки в модуле «Считать письма», но доступен только один протокол – IMAP. При выборе в поле «Настройки» необходимой почтовой службы, поля «Хост», «Порт» и «SSL» заполняются автоматически, при пользовательских настройках их необходимо заполнять вручную.

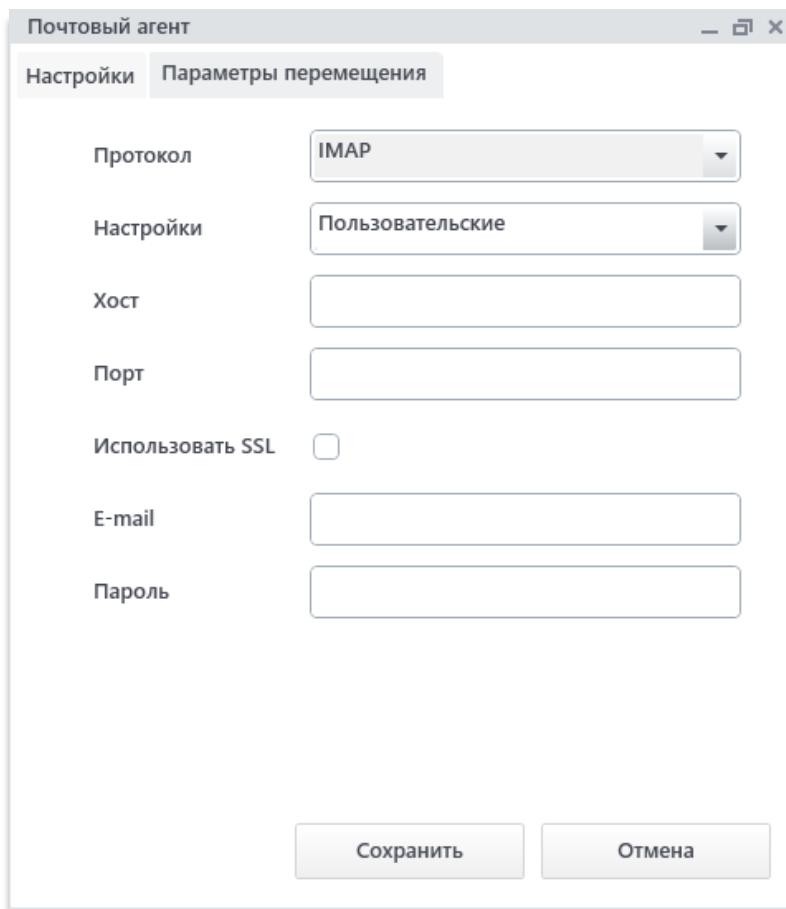


Рис. 7.22.1. Вкладка «Настройки» модуля «Переместить письмо»

Вкладка «Параметры перемещения»

На данной вкладке производится сама настройка переноса сообщения. Работает только для *одного сообщения*, нельзя в одном модуле перенести сразу несколько сообщений.

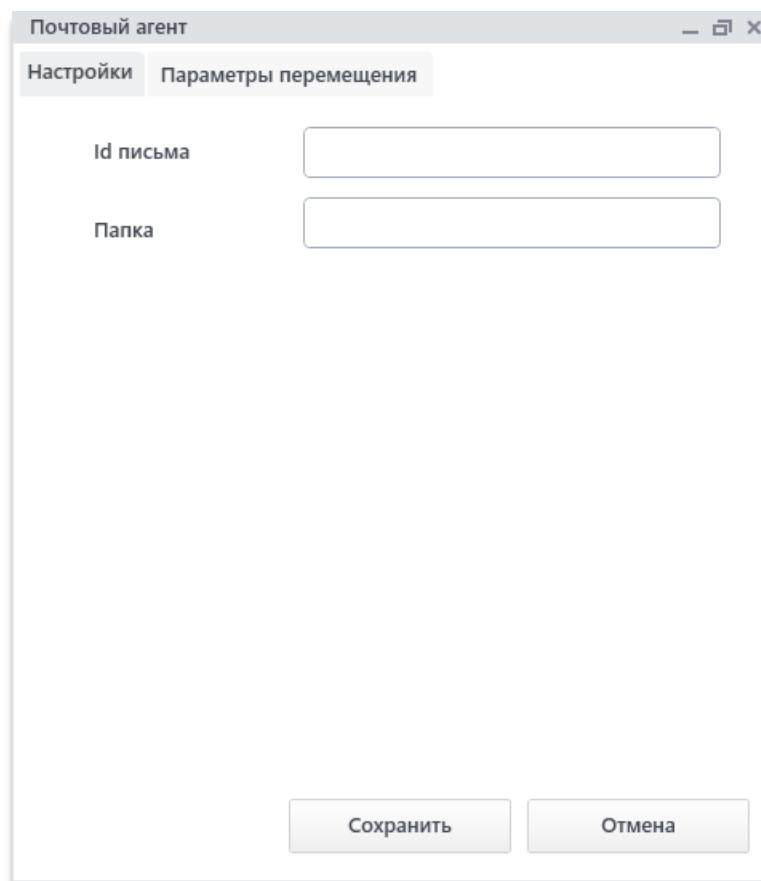


Рис. 7.22.2. Вкладка «Параметры перемещения»

В поле «**Id письма**» вносится id перемещаемого письма, который может быть получен с помощью модуля «Считать письма».

В поле «**Папка**» указывается название папки на почте, в которую должно быть перемещено письмо.

ЧАСТЬ VIII. МОДУЛЬ «РАБОТА С БРАУЗЕРОМ»

Глава 23. Основные моменты модуля

Модуль «Работа с браузером» построен на базе Selenium WebDriver⁶ и позволяет реализовывать основные методы работы со страницей браузера – открыть вкладку, перейти на вкладку, найти элемент на странице по её ключу и как-то его обработать.

Интерфейс модуля

Окно модуля делится на блок настройки прокси и выбор драйвера (пока доступен только Chrome), область создания команды для работы с браузером, группу кнопок управления действиями и таблицу, содержащую список созданных команд. Область создания команды состоит из нескольких полей для заполнения, причем, количество доступных для редактирования полей меняется в зависимости от выбранного действия в поле «Действие». Группа кнопок состоит из трех кнопок – «Добавить», «Редактировать» и «Удалить».

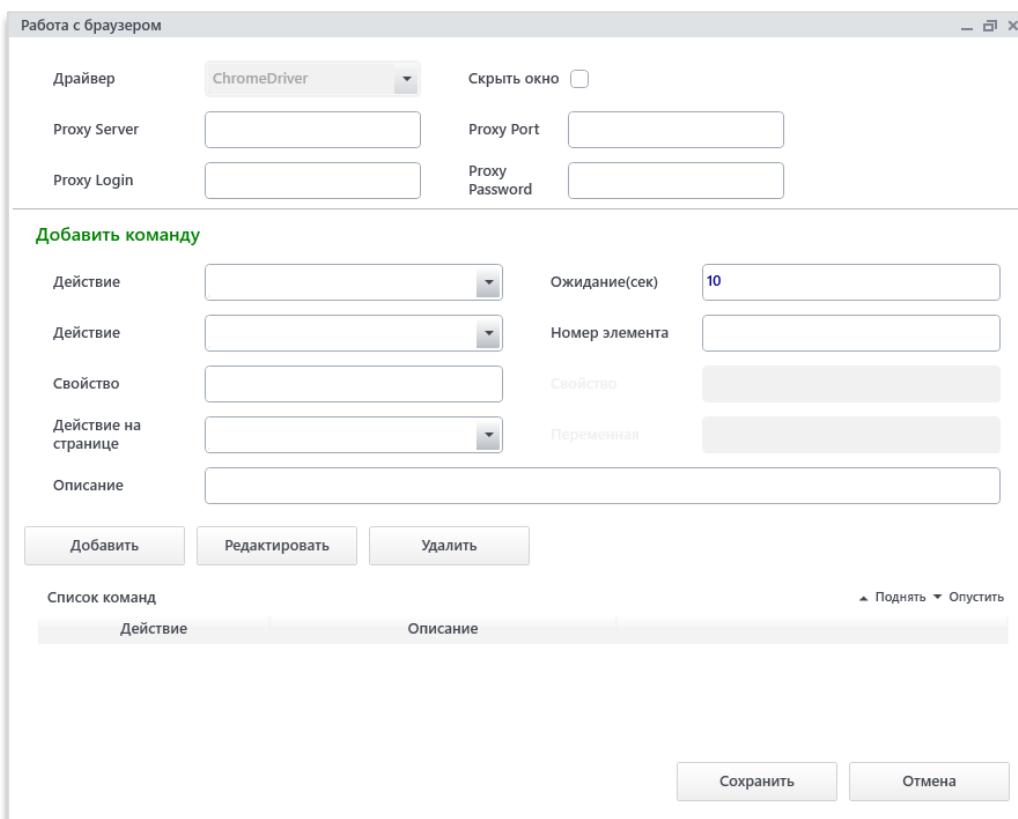


Рис. 8.23.1. Окно модуля «Работа с браузером»

Создание команды начинается с выбора действия. После его выбора список доступных полей в окне может измениться. Необходимо заполнить поля, требующиеся для выбранного

⁶ Selenium WebDriver – это программная библиотека для управления браузерами.

действия и нажать на кнопку «Добавить». Созданное действие отобразится в таблице «Список команд». Для редактирования команды следует выбрать её в списке команд, изменить необходимые поля и нажать «Редактировать». Если при этом нажать на кнопку «Добавить», то будет добавлено новое действие, похожее на то, что было выбрано в списке команд, но с внесенными изменениями. Можно ничего не менять и нажать на кнопку «Добавить», тогда получится полное копирование команды.

Для удаления ненужных команд необходимо выбрать команду в таблице и нажать «Удалить».

Робот будет выполнять все указанные команды последовательно, поэтому для изменения порядка команд в таблице можно воспользоваться кнопками в правом верхнем углу таблицы. Делается это следующим образом – выбирается строка, положение которой необходимо изменить, а затем нажимаются кнопки «Поднять» или «Опустить», в зависимости от того, где эта строка должна находиться.

Список команд		
Драйвер	Действие	Описание
		▲ Поднять ▼ Опустить

Рис. 8.23.2. Кнопки для изменения порядка команд

Для всех действий некоторые поля остаются неизменными, это все поля для настройки прокси, «Драйвер», «Действие» и «Примечание».

Поле «Драйвер» на данный момент недоступно для редактирования и в нем по умолчанию выбран драйвер для браузера Chrome, но в обозримом будущем этот список планируется дополнять другими браузерами.

Поля «Proxy Server» - «Proxy Password» предназначены для ввода настроек прокси браузера, если это необходимо.

Поле «Действие» — это выпадающий список с перечнем возможных действий этого модуля. Подробнее о них будет рассказываться в следующей главе.

Поле «Примечание» — строка с примечанием к команде.

Глава 24. Описание действий

Модуль предлагает использование 7 основных действий – перейти на страницу, перезагрузить страницу, перейти назад или вперед, работа со страницей, работа со вкладками и закрыть браузер. Каждое действие требует заполнения определенных полей.

Обратите внимание, что модуль создает свой экземпляр браузера, без авторизованных аккаунтов и прочего. Созданный браузер закрывается сам в конце работы робота.

Перечень и описание действий

Действие «Перейти»

С помощью действия «Перейти» происходит переход на заданную страницу. В области создания команды открывается поле – «URL».

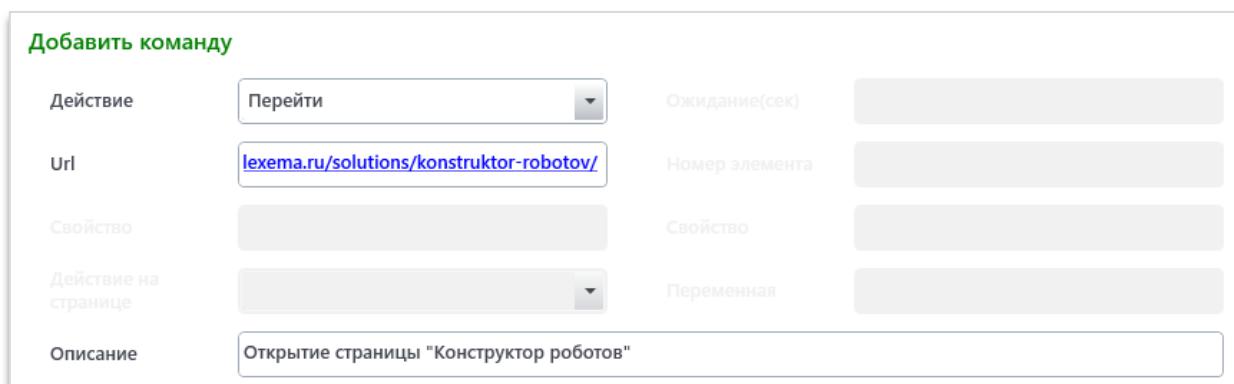


Рис. 8.24.1. Пример настройки действия «Перейти»

Если уже открыта какая-либо страница, то браузер перейдет от неё к странице, указанной в поле «URL», то есть исходная страница будет заменена новой. Если браузер еще не был запущен, то он будет открыт с этой страницы.

Действие «Перезагрузить страницу»

Действие позволяет перезагружать уже открытую страницу. Перед использованием данного действия, убедитесь, что в браузере есть открытая страница.

Действие не имеет никаких дополнительных для настройки полей.

Действие «Назад»

Действие позволяет выполнять переход к странице, которая была открыта в браузере до текущей.

Действие «Вперед»

Действие позволяет выполнять переход к странице, которая была открыта в браузере после текущей, если таковая была.

Действие «Работа со страницей»

Данное действие предоставляет возможность работать с отдельным элементом страницы. Для настройки действия появляются следующие поля: «Тип поиска элемента», «Строка поиска», «Действие на странице», «Ожидание» и «Номер элемента». Еще два поля – «Свойство» и «Переменная» становятся доступными в зависимости от выбора действия на странице.

Добавить команду

Действие	<input type="button" value="▼"/>	Ожидание(сек)	<input type="text" value="10"/>
Тип поиска элемента	<input type="button" value="▼"/>	Номер элемента	<input type="text"/>
Строка поиска	<input type="text"/>	Свойство	<input type="text"/>
Действие на странице	<input type="button" value="▼"/>	Переменная	<input type="text"/>
Описание	<input type="text"/>		

Рис. 8.24.2. Поля для настройки действия «Работа со страницей»

Поле «Тип поиска элемента» представляет собой выпадающий список с перечнем методов поиска элемента на странице, используется в паре с полем **«Строка поиска»**.

Ниже будут перечислены все типы поиска элемента с примерами заполнения поля **«Строка поиска»**:

- GetElementsByTagName – поиск элементов по их свойству name.

```
<div name="element">
  <p>some content</p>
</div>
```

Листинг 8.24.1. Пример элемента со свойством name

«Строка поиска» для нахождения элемента div - element.

- GetElementsById – поиск элементов по их свойству id.

```
<div id="element">
  <p>some content</p>
</div>
```

Листинг 8.24.2. Пример элемента со свойством id.

«Строка поиска» для нахождения элемента div - element.

- GetElementsByClass – поиск элементов по их свойству class.

```
<div id="element">
  <p>some content</p>
</div>
```

Листинг 8.24.3. Пример элемента со свойством class

«Строка поиска» для нахождения элемента div - myClass.

- GetElementsByCssSelector – поиск элементов по их CSS селектору.⁷

```
<div class="main">
  <p>text</p>
  <p> Another text </p>
</div>
```

Листинг 8.24.4. Пример элемента div со свойством class

«Строка поиска» для нахождения элемента div - div.main.

- GetElementByLinkText – поиск элементов-ссылок по их тексту. Ищутся только элементы с полным совпадением.

```
<div>
  <a ref=" ...">text</a>
  <a ref=" ...">Another text</a>
</div>
```

Листинг 8.24.5. Пример элементов – ссылок a

«Строка поиска» нахождения первого элемента a на примере – text.

- GetElementsByPartialLinkText - поиск элементов-ссылок по их тексту. Ищутся все элементы, содержащие заданную строку.

```
<div>
  <a ref=" ...">text</a>
  <a ref=" ...">Another text</a>
</div>
```

Листинг 8.24.6. Пример элементов – ссылок a

⁷ Селектор — это часть CSS-правила, которая сообщает браузеру, к какому элементу (или элементам) веб-страницы будет применён стиль.

«Строка поиска» нахождения двух элементов а – text

- GetElementsByTagName – поиск элементов по их тегу⁸.

```
<div>
  <a ref=" ...>text</a>
  <p>Another text</p>
</div>
```

Листинг 8.24.7. Пример элементов с разными тегами

«Строка поиска» нахождения элемента а – a.

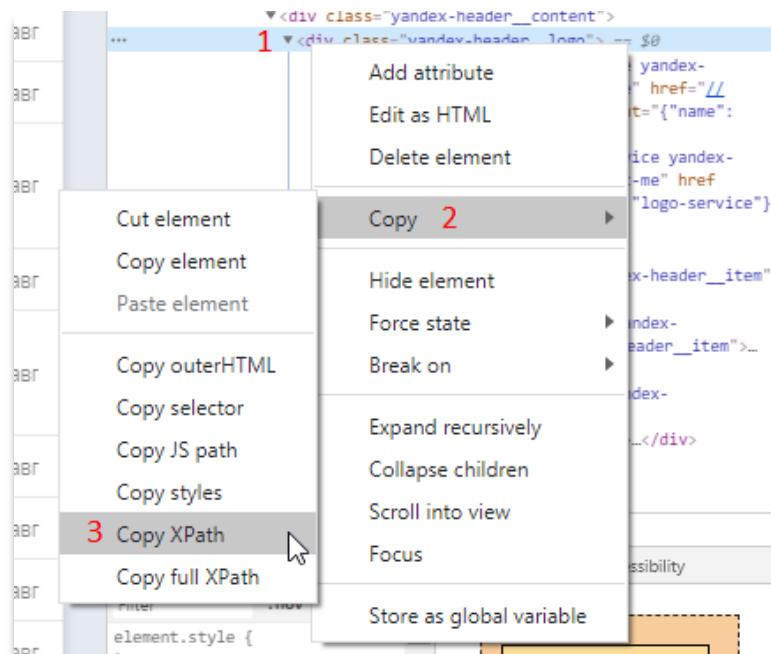
- GetElementsByTagName – поиск элементов по языку запросов HTML – xpath.

```
<div class="main">
  <p>text</p>
  <p>Another text </p>
</div>
```

Листинг 8.24.8. Пример элементов div со свойством class

«Строка поиска» для нахождения элемента div - //div[@class='main'].

Чтобы получить строку xpath для необходимого вам элемента в консоли разработчика в браузере, нужно нажать по этому элементу правой кнопкой мыши в коде – копировать – XPath:



⁸ Тег – элемент языка разметки.

Рисунок 8.24.3. Получение строки *xpath* из консоли браузера

Поле «Действие на странице» предлагает выбор действия из выпадающего списка, которое необходимо произвести над найденными элементами. Доступны следующие действия:

- Записать – вводит заданный в поле «Переменная» текст (например, после нажатия на определенный элемент для ввода текста);
- Получить text – получает значение свойства «text» всех найденных или только выбранного элемента и возвращает результат в виде списка в переменную;
- Отправить – нажатие клавиши «ВВОД» («ENTER») – применение записанных данных (например, в строке поиска можно ввести какой-либо текст действием «Записать», а затем «Отправить» его);
- Нажать – нажатие левой кнопкой мыши на заданный элемент (обязательно заполнение поля «Номер элемента»);
- Навести мышь – наведение мыши на конкретно заданный элемент (обязательно заполнение поля «Номер элемента»);
- Получить Property – при выборе действия появляется поле «Название Property» - возвращает значение указанного в поле «Название Property» свойства всех найденных элементов или конкретно заданного. Результатом будет являться список;
- Получить CSS свойство – при выборе действия появляется поле «Свойство CSS» - возвращает значение указанного в поле «Свойство CSS» свойства всех найденных элементов или конкретно заданного. Результатом будет являться список;
- Получить Attribute – при выборе действия появляется поле «Свойство Attribute» - возвращает значение указанного в поле «Свойство Attribute» свойства всех найденных элементов или конкретно заданного. Результатом будет являться список;
- Получить элементы – возвращает список указанных элементов для дальнейшей работы с ними;
- Выполнить JavaScript – данное действие позволяет выполнить JavaScript код в браузере для получения некоторой информации. Для записи результата выполнения кода в переменную, необходимо в начале кода добавлять «return», например «return document.querySelectorAll("[height='24']").length» (такой запрос вернет количество элементов на странице, высота которых равна 24);
- Нажать кнопку на клавиатуре – позволяет нажать кнопку на клавиатуре внутри браузера.

В поле «Переменная» вводится название переменной, начиная с символов «v.», в которую будет помещен результат выполнения команды или, наоборот, из которой будут братьсяся значения для её выполнения.

Поле «Ожидание (сек)». В поле вводится количество секунд, которые необходимо подождать после выполнения команды.

Поле «Номер элемента» предназначено для указания индекса найденного элемента, к которому необходимо применить действие.

Пример заполненного действия:

Добавить команду

Действие	Работа со страницей	Ожидание(сек)	10
Тип поиска элемента	GetElementByName	Номер элемента	0
Строка поиска	q	Свойство	
Действие на странице	Записать	Переменная	Lexema-RPA
Описание	Вставка данных в поисковик		

Рис. 8.24.3. Пример настройки действия «Работа со страницей»

На Рис. 8.24.3 представлен пример действия, настроенного для поиска элемента, имя которого равно «q». После того как элемент найден, в него будет вставлен текст «поиск».

Действие «Работа с вкладками»

Данное действие позволяет манипулировать вкладками в окне браузера. Для настройки действия открывается как минимум одно поле – «Действие», остальные открываются в зависимости от выбранного в нем действия.

Добавить команду

Действие	Работа с вкладками	Ожидание(сек)	
Действие на вкладке		Номер элемента	
Строка поиска		Свойство	
Действие на странице		Переменная	
Описание			

Рис. 8.24.4. Поля для настройки действия «Работа со вкладками»

Доступные действия для работы с вкладками:

- Добавить вкладку – создаст новую вкладку, и та станет активной;
- Закрыть вкладку – закроет активную вкладку;

- Перейти на вкладку – при выборе действия появляется поле «Номер вкладки» - переходит на вкладку, номер которой указан в поле «Номер вкладки» (начиная с 0);
- Получить количество вкладок – при выборе действия появляется поле «Переменная» - возвращает в указанную переменную количество вкладок в браузере.

Действие «Закрыть браузер»

Закрывает браузер, открытый с помощью модуля, не требует дополнительных настроек.

Часть IX. МОДУЛЬ «ИНТЕРФЕЙС»

Глава 25. Основные моменты

Данный модуль предназначен для создания диалоговых окон для общения с пользователем робота. Это могут быть окна, позволяющие прикрепить файл, ввести какие-либо данные – дату, имена, почтовые адреса и другое, с чем впоследствии будет работать робот, или наоборот, окна, отображающие результат работы.

Создание окна заключается в добавлении на него контроллов – элементов интерфейса таких, как надписи, текстовые поля, кнопки, выпадающие списки.

Основное окно

Окно модуля состоит из нескольких частей:

- Настройки окна – общие настройки диалогового окна;
- Параметры элемента интерфейса - настройка добавляемого элемента;
- Список элементов – таблица, в которой содержатся все добавленные элементы;
- Окно с предварительным просмотром диалогового окна на текущий момент (правая часть окна).

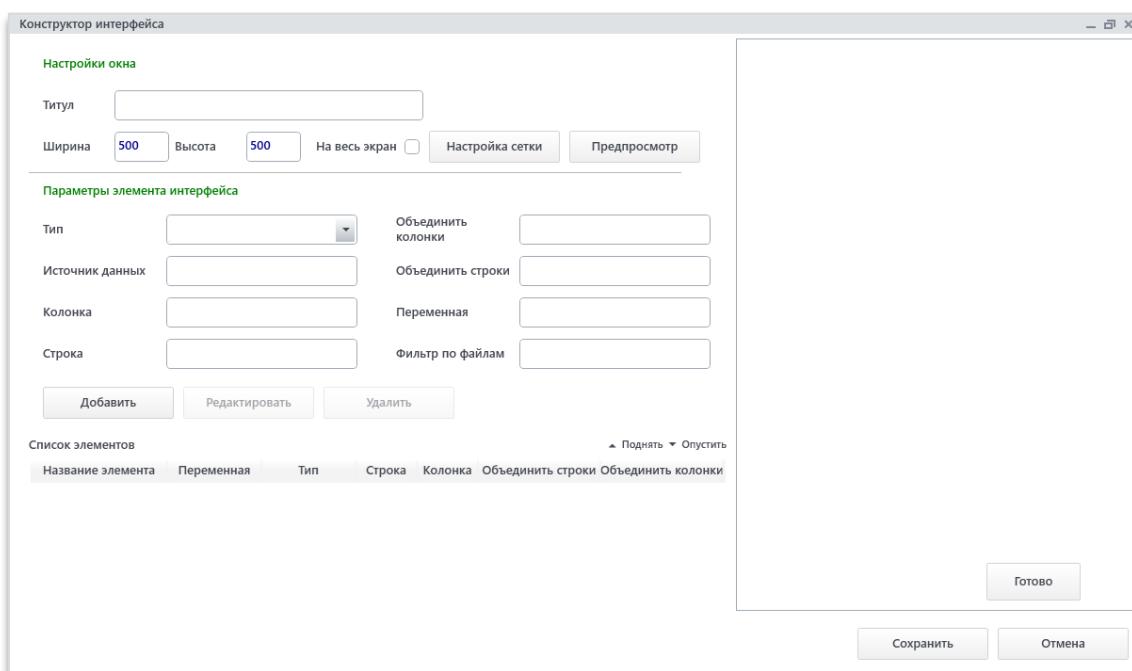


Рис. 9.25.1. Основное окно модуля

Начинать работу с диалоговым окном следует с общих настроек, а именно с настройки сетки, которую можно открыть по кнопке «Настройка сетки».

Глава 26. Настройки окна

К настройкам окна относится название окна – «Титул», его ширина и высота и настройка сетки. При установке флажка «На весь экран» поля «Высота» и «Ширина» не будут учитываться.

По нажатию на кнопку «Настройка сетки» появится окно настроек строк и колонок сетки.

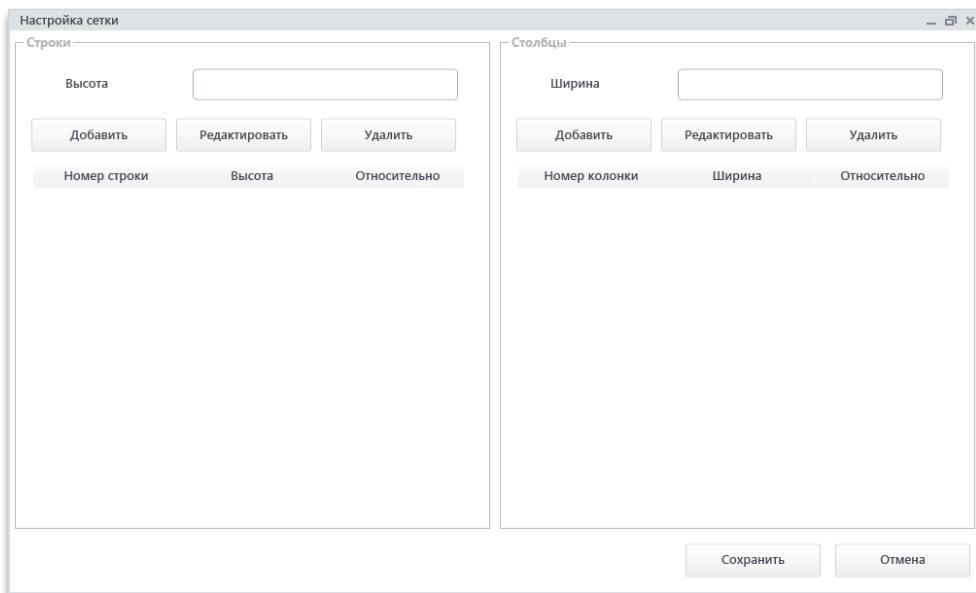


Рис. 9.26.1. Настройка сетки

Сетка – это условная разметка элементов интерфейса, некоторая таблица, в которую будут помещаться контролы. Например, если нужно создать окно, в которое пользователь бы вводил два каких-то параметра – конечную и начальную даты, за которые необходимо скачивать почту, то настроить диалоговое окно для ввода этих дат можно несколькими способами:

- Первый способ

- в две строки, 4 элемента интерфейса – на первой строке подпись и поле для ввода начальной даты, на второй строке подпись и поле для ввода конечной даты

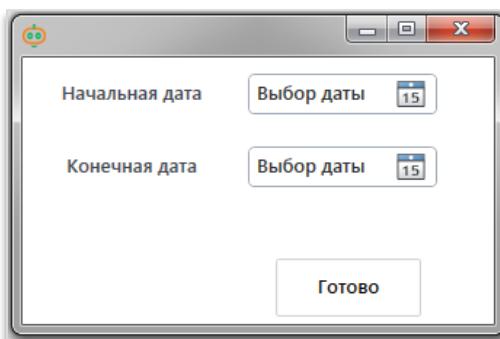


Рис. 9.26.2. Пример окна №1

Для такой настройки сетки необходимо задать две колонки – первая, для надписей, вторая для полей ввода, и две строки – первая для начальной даты, вторая – для конечной.

- Второй способ

- в одну строку, 3 элемента интерфейса – подпись, поле для ввода первой даты, поле для ввода второй даты.

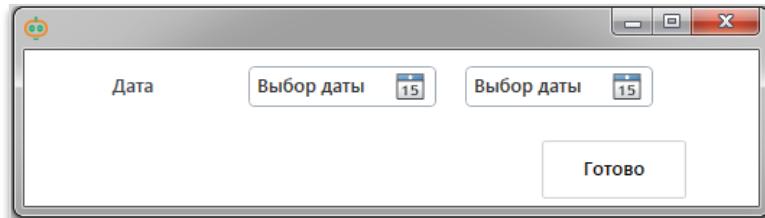


Рис. 9.26.3. Пример окна №2

В таком случае используется три колонки и одна строка.

- Третий способ

- в одну строку, 4 элемента интерфейса – два поля для ввода и подписи к каждому из них.

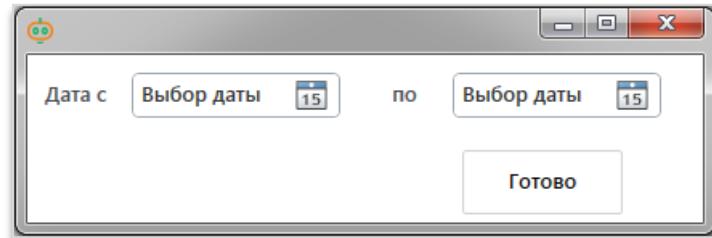


Рис. 9.26.4. Пример окна №3

Здесь используются 4 колонки и 1 строка.

Для каждой колонки можно задать свою ширину, для каждой строки – высоту:

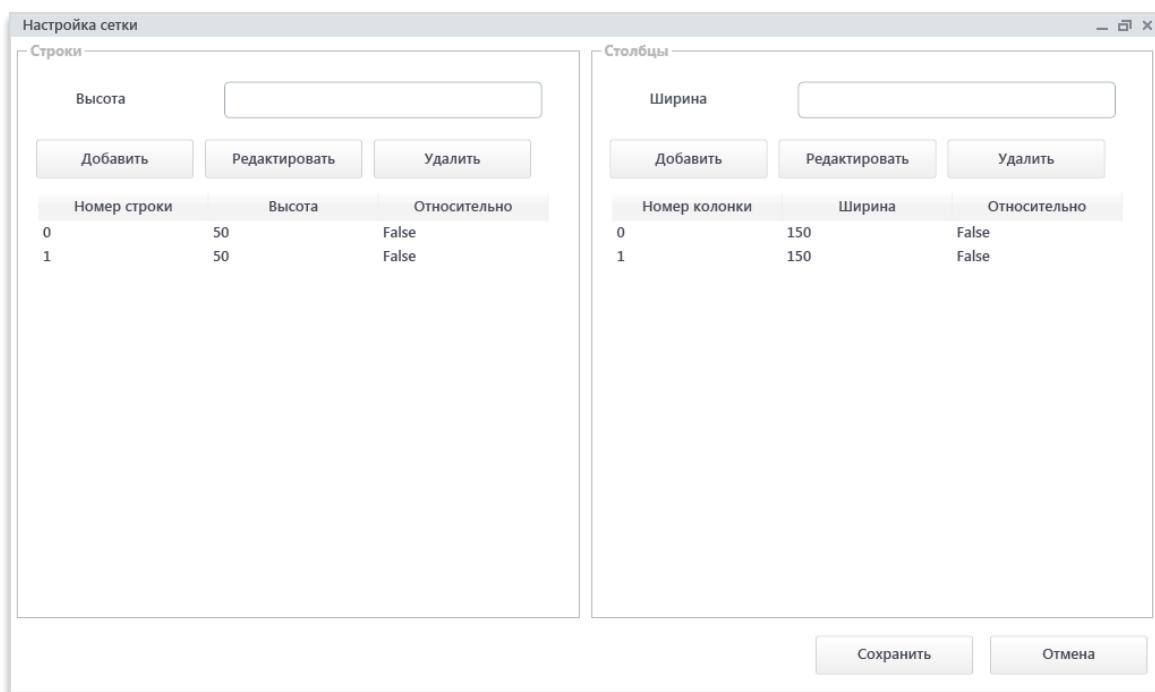


Рис. 9.26.5. Пример настройки сетки

Высота и ширина вводятся в пикселях, но можно использовать относительные размеры окна, например, если одна колонка должна быть больше другой в два раза и их всего две, то в поле «Ширина» можно ввести «1*» для первой колонки, и «2*» для второй:

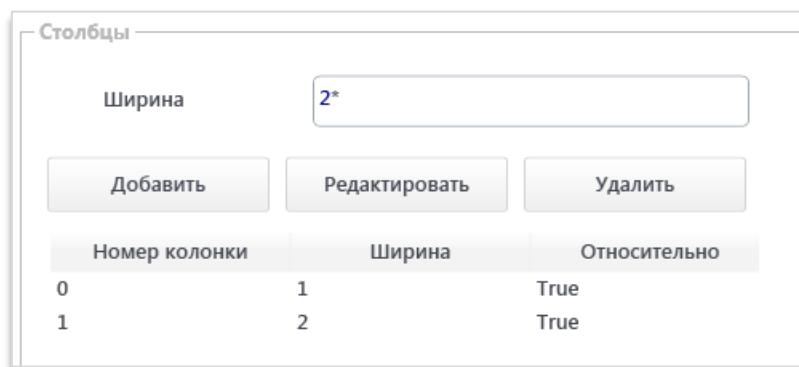


Рис. 9.26.6. Пример использования относительных размеров

Результат такой разметки:



Рис. 9.26.7. Пример разметки

Нумерация строк и колонок начинается с 0. С помощью кнопки «Удалить» можно удалить ненужную строку или колонку, по кнопке «Редактировать» можно изменить высоту или ширину.

После настройки сетки она будет отображена пунктирной линией в окне предварительного просмотра в правой части окна, и затем можно приступить к добавлению контроллов.

Глава 27. Добавление элемента интерфейса

Для добавления на окно нового элемента необходимо заполнить поля, представленные на Рис. 9.27.1:

Параметры элемента интерфейса	
Тип	<input type="button" value="▼"/>
Источник данных	<input type="button"/>
Колонка	<input type="button"/>
Строка	<input type="button"/>
Объединить колонки	<input type="button"/>
Объединить строки	<input type="button"/>
Переменная	<input type="button"/>
Фильтр по файлам	<input type="button"/>

Рис. 9.27.1. Поля для добавления нового элемента

Обязательными полями являются все, кроме полей «Объединить колонки/строки». Поле «Фильтр по файлам» доступно только для элемента «Выбор файла».

Поле «Тип»

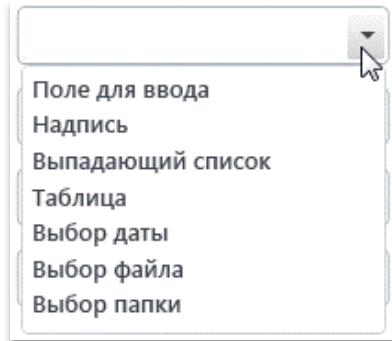


Рис. 9.27.2. Выпадающий список с выбором типа элемента

Сначала выбирается тип добавляемого элемента интерфейса из предложенных:

- Поле для ввода – поле, в которое можно ввести любую строку;

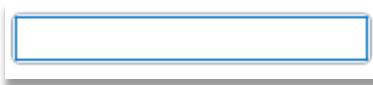


Рис. 9.27.3. Элемент «Поле для ввода»

- Надпись – статичная строка текста, которая не может быть отредактирована пользователем;

Надпись

Рис. 9.27.4. Элемент «Надпись»

- Выпадающий список – поле со стрелочкой сбоку, по нажатию которого открывается список заранее определенных значений;

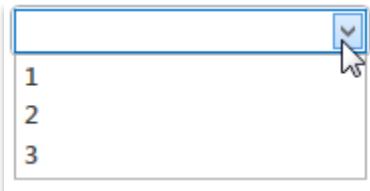


Рис. 9.27.5. Элемент «Выпадающий список»

- Таблица – отображение в виде таблицы заранее созданной переменной. Возможно редактирование полей таблицы, но невозможно создание новых колонок в ней;

Column1	Column2	Column3
1.1 ячейка	1.2 ячейка	1.3 ячейка
2.1 ячейка	2.2 ячейка	

Рис. 9.27.6. Элемент «Таблица»

- Выбор даты – поле с иконкой календаря, по нажатию на которую пользователю будет предложено выбрать определенную дату. Возможен ввод значения даты вручную, при вводе других строк поле автоматически подберет наиболее подходящую дату к введенным данным. Формат даты ДД.ММ.ГГГГ;

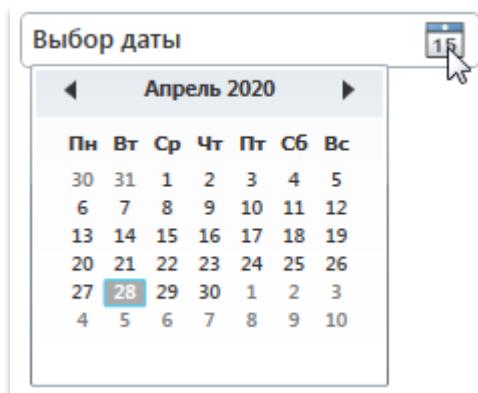


Рис. 9.27.7. Элемент «Выбор даты»

- Выбор файла – поле с иконкой в виде трех точек, по нажатию на которую пользователю будет предложено выбрать файл. Путь до выбранного файла запишется в это поле.



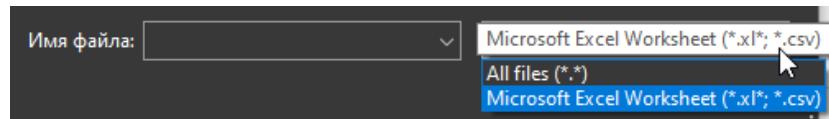
Рис. 9.27.8. Элемент «Выбор файла»

Для фильтрации отображаемых пользователю файлов можно воспользоваться полем «Фильтр по файлам», в которое вводится строка вида: «Название типа файлов (доступные расширения)|доступные расширения». Можно задать несколько настроек фильтрации, разделителем для этого является символ «|». Пример:

All files (*.*)|*.|Microsoft Excel Worksheet (*.xl*; *.csv)|*.xl*; *.csv

В данном случае отображается две настройки фильтрации:

1. All files (*.*), в которой отображаются файлы с любым расширением;
2. Microsoft Excel Worksheet (*.xl*; *.csv), в которой отображаются файлы с расширениями .csv и начинающимися на .xl (то есть .xlsx, xls и другие).



- Выбор папки – поле с иконкой в виде трех точек, по нажатию на которую пользователю будет предложено выбрать папку. Путь до выбранной папки запишется в это поле. Выглядит так же, как и «Выбор файла».

Поле «Источник данных»

В это поле вводится название переменной или строка, на основе которой контрол будет заполнен. Немного поподробнее о поле при выборе различных элементов интерфейса:

- Для типа элемента «Надпись» туда вводится значение, которое должно быть отображено в выводимой надписи;
- Для «Поле для ввода», «Выбор даты», «Выбор файла» и «Выбор файла» вводится значение, которое сразу будет указано в поле, если это необходимо. Если поле должно быть пустым, то не надо заполнять «Источник данных»;
- Для «Таблицы» необходимо подать заполненную табличную переменную для её отображения;
- Для элемента интерфейса «Выпадающий список» необходимо передать набор значений, которые пользователю будет предложено выбрать. Делается это через символ «;». Это могут быть строки, переменные-элементы, переменные-справки или табличные переменные, состоящие из одного столбца. Например, введено следующее значение, при приведенных далее примерах заполнения переменных:

Тип	Выпадающий список
Источник данных	EUR;v.usd;v.currency

Рис. 9.27.9. Пример заполнения поля «Источник данных»

Редактирование переменной

Название переменной	usd
Тип переменной	Элемент
Значение	USD

Сохранить Отмена

Переменная

Название переменной	currency
Тип переменной	Список
Значение	<input type="button" value="Добавить"/>
Значение	RUB GBP CHF

Сохранить Отмена

Рис. 9.27.10. Примеры заполнения используемых переменных

Результат будет выглядеть следующим образом:

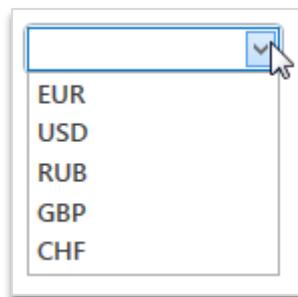


Рис. 9.27.11. Пример результирующих данных

Поля «Колонка» и «Строка»

В эти поля необходимо ввести номер колонки и строки сетки (таблицы), в которых должен располагаться добавляемый контрол. Например, в примере с датами (Рис. 9.26.2) контрол «Надпись» - «Начальная дата» располагается в 0 колонке и 0 строке, а надпись «Конечная дата» - в 0 колонке и 1 строке.

Поля «Объединить колонки» и «Объединить строки»

В некоторых случаях необходимо выводить поля не в конкретной клеточке сетки, а в объединении нескольких клеток. Например – нужно предложить пользователю выбрать Excel-файл для записи в него результатов и ввести начальную и конечную дату, по которым будут собираться некоторые данные. Хотелось бы объединить это в две строки, но для ввода даты нужно как минимум 2 поля, а поле для выбора файла – одно.

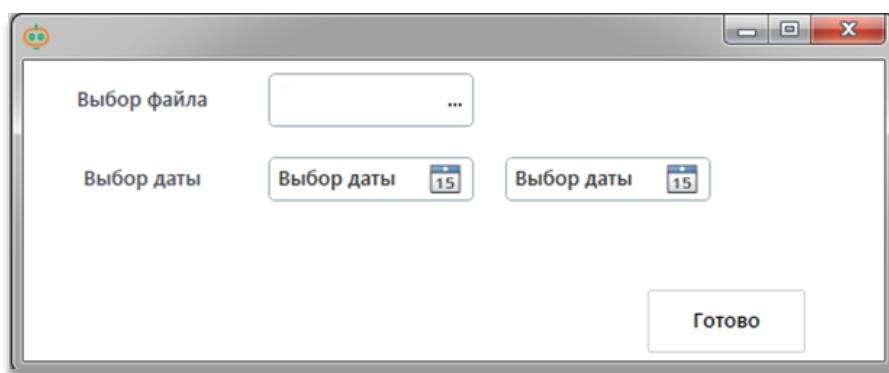


Рис. 9.27.12. Пример окна без объединения колонок

Для того, чтобы растянуть поле для выбора файла по длине полей для ввода даты, нам необходимо разместить его не просто в 1 колонке и 0 строке, но и объединить для него две колонки, чтобы оно занимало сразу обе:

Параметры элемента интерфейса

Тип	Выбор файла	Объединить колонки	2
Источник данных	v.file	Объединить строки	
Колонка	1	Переменная	v.file
Строка	0		

Рис. 9.27.13. Пример заполнения полей элемента интерфейса для настройки объединения колонок

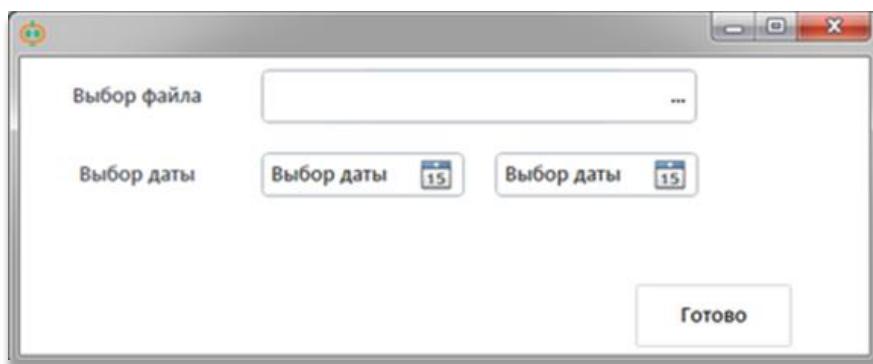


Рис. 9.27.14. Пример окна с объединенными колонками в первой строке сетки

То есть при заполнении поля «Объединить колонки» колонки, начиная с той, что указана в поле «Колонка», будут объединены по горизонтали в одну.

Аналогично с полем «Объединять строки» - указанное количество строк, начиная с той, что указана в поле «Строка», будут объединены в одну:

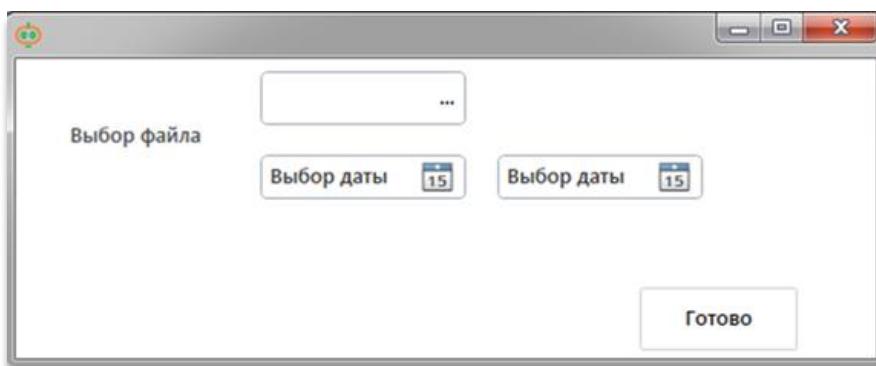


Рис. 9.27.15. Пример окна с объединенными строками в первой колонке сетки

Поле «Переменная»

В это поле вводится название переменной, в которую будет записано значение, введенное (для выпадающего списка – выбранное) пользователем. Является обязательным полем для всех типов контроллов, кроме «Надписи».

После заполнения всех полей необходимо нажать на кнопку «Добавить». После нажатия на нее элемент сразу отобразится в окне предпросмотра в правой части окна.

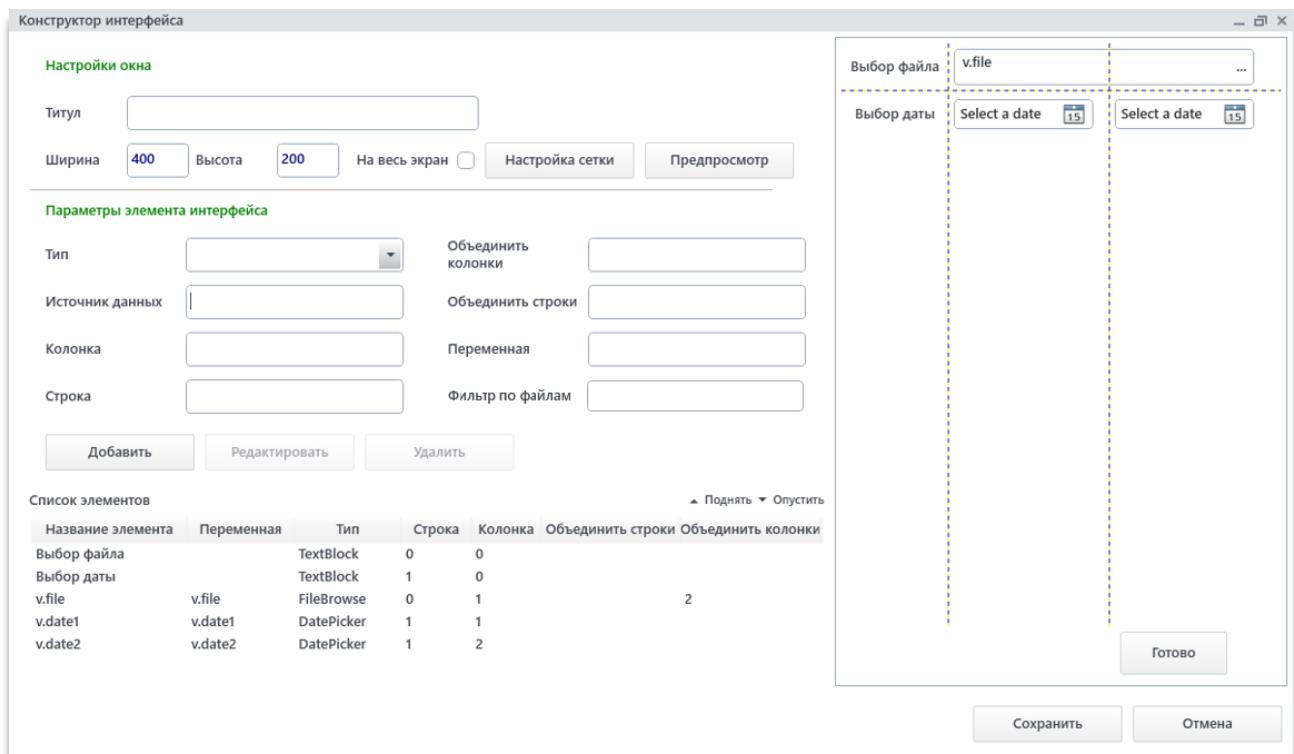


Рис. 9.27.16. Пример заполненного окна модуля «Интерфейс»

По кнопке «Предпросмотр» откроется диалоговое окно в таком виде, в котором оно будет представлено пользователю.

Удалить или отредактировать элемент можно, предварительно выбрав его в области «Список элементов», а затем нажав на соответствующую кнопку, находящуюся над списком с элементами.

После добавления всех необходимых контроллов нужно нажать кнопку «Сохранить» в правой нижней части окна.

ЧАСТЬ X. МОДУЛИ ДЛЯ РАБОТЫ С ТЕКСТОМ

Глава 28. Модуль «Обработка текста»

Модуль «Обработка текста» предлагает несколько методов сравнения текста. В некоторых задачах требуется знание о том, как сильно различен текст между эталонным и указанным. Это задачи, относящиеся к компьютерной лингвистике и искусственному интеллекту.

Интерфейс модуля

Окно модуля состоит из части составления команды, кнопок управления командами и списком команд в виде таблицы. Часть составления команды состоит из следующих полей: выпадающий список «Действие» с доступным набором методов анализа текста, два поля «Строка 1» и «Строка 2», предназначенные для ввода двух строк текста или переменных, содержащих текст, и поле «Переменная результат» - для названия переменной, в которую будет помещен результат работы модуля.

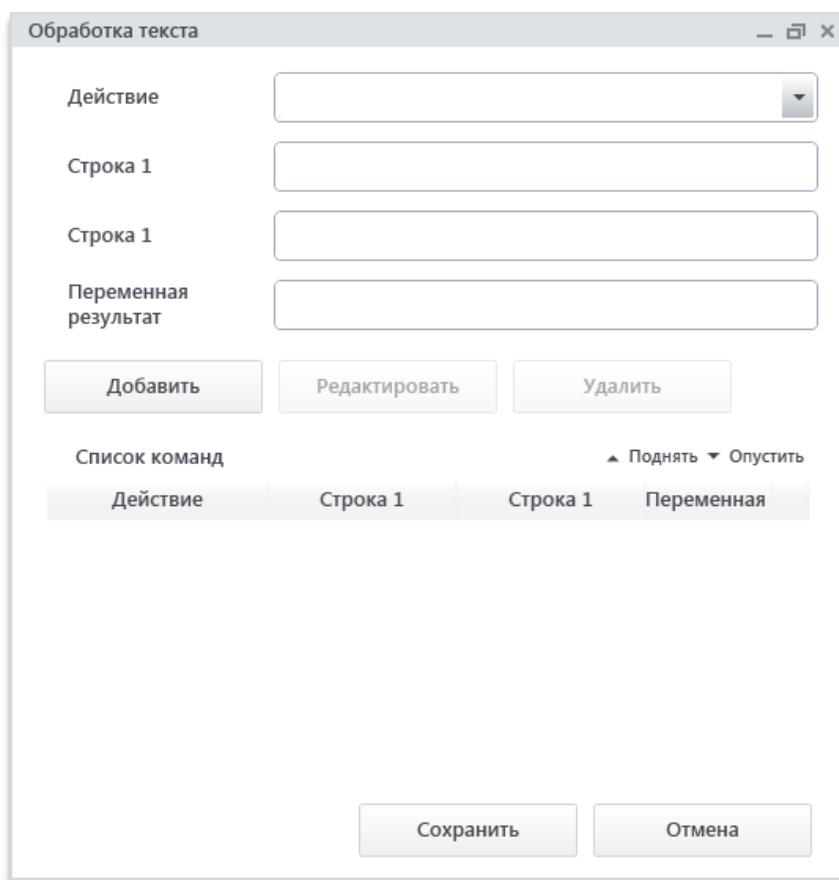


Рис. 10.28.1. Окно модуля

По кнопке «Добавить» созданная команда поместится в таблицу «Список команд». Для редактирования команды из списка нужно выбрать её в таблице, изменить необходимые поля

и нажать на кнопку «Редактировать». Для удаления команды нужно выбрать ее в списке и нажать на кнопку «Удалить». С помощью стрелочек в углу таблицы можно менять команды местами.

Методы анализа текста

В модуле предложены следующие методы анализа текста:

1. **Расстояние Левенштейна** – рассчитывает разницу между двумя строками. Например – «Lexema RPA» и «Lexema SR» отличаются на 3 символа – слова «Lexema» совпадают полностью, остальные символы различны, то есть результат, записанный в переменную, будет равен 3;
2. **3-граммы** – метод, основанный на работе с n-граммами, в нашем случае n=3 – оценивается схожесть каждого 3 символов. Чем больше число (до 1), тем большую схожесть имеют строки. В примере «Lexema RPA» и «Lexema SR» результатом будет число 0,52.
3. **Сходство Джаро-Винклера** – мера схожести строк для измерения расстояния между двумя последовательностями символов. Чем меньше расстояние Джаро-Винклера для двух строк, тем больше сходства имеют эти строки друг с другом. Для примера «Lexema RPA» и «Lexema SR» результатом будет число 0,5.

Глава 29. Модуль «Распознавание текста»

Модуль «Распознавание текста» предназначен для считывания текста с указанного изображения и представления полученного текста в виде объекта.

Интерфейс модуля

Модуль состоит из двух полей – «Путь к файлу» и «Переменная», и флажка «Скриншот».

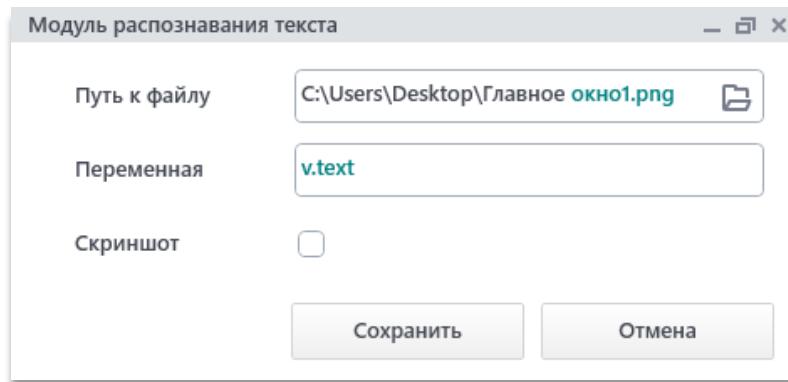


Рис. 10.29.1. Окно модуля

Поле «Путь к файлу». Заполняется путем до файла, который необходимо распознать. Это может быть картинка или PDF-документ.

Поле «Переменная». Заполняется названием переменной, начиная с символов «v.», в которую будет помещен результат распознавания.

Флажок «Скриншот». Флажок ставится, когда необходимо распознать скриншот экрана в момент работы данного модуля в процессе выполнения робота. В таком случае поле «Путь к файлу» указывать не нужно.

Структура результирующего объекта – массив листов документа, каждый элемент которого – объект, состоящий из двух полей – number и words. Поле number содержит в себе номер листа документа (с 1-цы), поле words представляет собой массив объектов всех слов со страницы. Структура объекта слова – value, x, y, page. В value содержится слово, в x и y координаты слова x и y соответственно, в page – номер страницы, на котором содержится слово.

ПРИМЕР.

Пусть имеется следующее изображение формата jpg. Подадим его в модуль распознавания текста.

Часть VIII

Модули работы с текстом

Рис. 10.29.2. Изображение, поданное на распознавание текста

Результат, записанный в переменную при распознавании изображения:

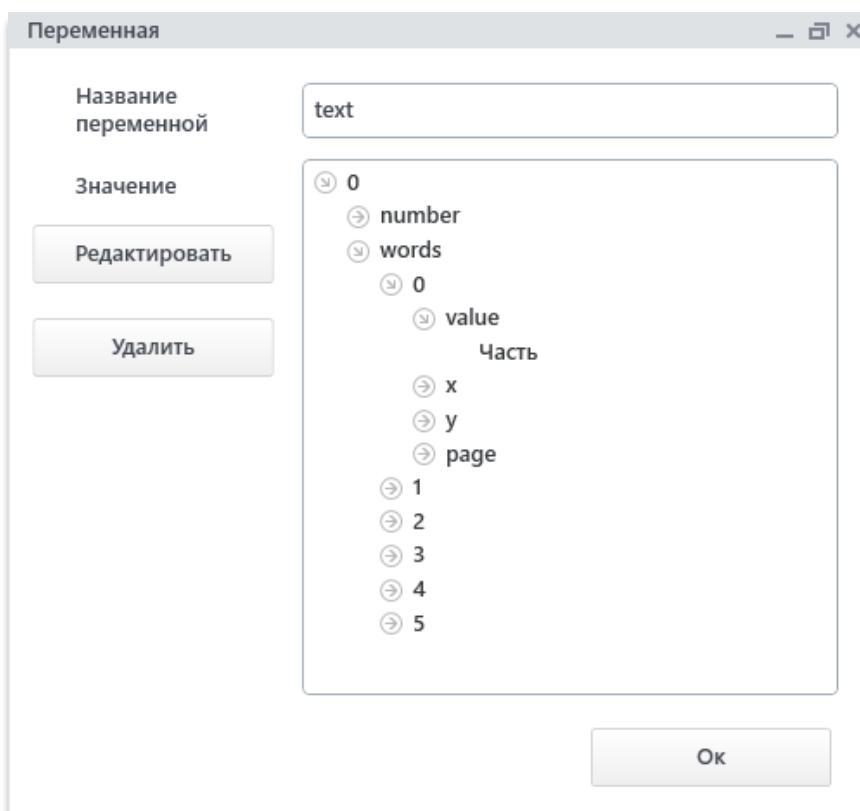


Рис. 10.29.3. Пример заполнения переменной

Результат содержит информацию о страницах, координате и значении каждого слова изображения.

Пример получения некоторого слова со второго листа документа: `v.text[1].words[100].value`, где `v.text` – переменная, в которую записан результат работы модуля, `[1]` – номер листа в документе (нумерация по переменным-спискам начинается с 0), `[100]` – 101-ое слово в файле, `value` – само значение слова.

Глава 30. Модуль «Распознавание текста ABBYY»

Модуль «Распознавание текста ABBYY» работает, в основном, с PDF-файлами. В отличие от предыдущего модуля имеет более высокую точность и скорость работы.

Результат распознавания записывается в переменную-объект, в которой можно обратиться к конкретной строке, таблице, слову.

Интерфейс модуля

Данный модуль состоит из двух полей – «Путь к PDF-файлу» и «Переменная».

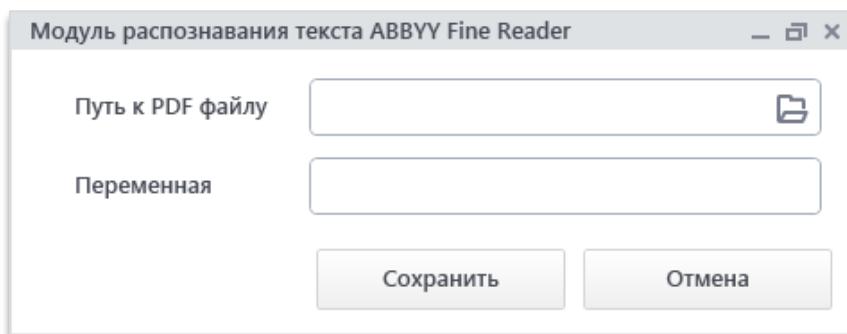


Рис. 10.30.1. Окно модуля

Поле «Путь к PDF файлу». В поле указывается путь до распознаваемого файла/картинки.

Поле «Переменная». В поле вводится имя переменной, начиная с символов «v.», в которую будет занесен результат распознавания.

PDF файл может состоять из нескольких страниц, поэтому переменная будет являться списком страниц. Например, `v.text[0]` – первая распознанная страница файла.

Весь распознанный текст делится на две части – основной текст и табличная часть.

Основной текст

Для обращения к тексту, необходимо обращаться к полю `rows_word[0]` (в данной записи `[0]` означает, что идет обращение к первой текстовой части, в других версиях ABBYY возможно использование большего разбиения текста). Затем указывается, как отдельное поле, номер строки и, через точку, номер слова в строке – `v.text[0].rows_word[0].2.4`. Затем, после выбора нужного слова, можно получить его значение, обратившись к полю `value` – `v.text[0].rows_word[0].2.4.value` – такая строка выдаст значение слова, находящегося на первой странице распознанного документа, во второй строке и является четвертой слева.

Табличная часть

Для обращения к таблице, необходимо обращаться к полю `tables[0]` (`[0]` нужен для корректной работы, его смысловая часть заложена в продукте ABBYY). Затем отдельным

полем вводится номер таблицы, так как на странице их может быть несколько – **v.text[0].tables[0].0**. После этого вводится номер ячейки в таблице в виде одного числа. Если вы хотите найти ячейку и знаете её номер колонки и строки, то можно получить индекс рассматриваемой ячейки переменной в виде «строка,колонка», обратившись к полю **index** – **v.text[0].tables[0].0.5.index** (5 – пятая ячейка в массиве распознанных ячеек). Для получения значения из ячейки нужно обратиться к её значению, а затем либо собрать всю строку, записанную в ячейку, либо обратиться к конкретному слову по его порядковому номеру – **v.text[0].tables[0].0.5.value.1.value** – значение второго слова из пятой ячейки первой таблицы с первой страницы распознанного текста.

Для получения количества ячеек в таблице используйте следующую конструкцию – **v.text[0].tables[0].0.length**, то есть используйся метод **«length»** после того поля, после которого идет поле, количество элементов которого необходимо узнать.

Для получения количества слов в ячейке используйте **v.text[0].tables[0].0.5.length**.

ЧАСТЬ XI. ДРУГИЕ МОДУЛИ

Глава 31. Модуль «Вычисления»

Модуль «Вычисления» является связующим модулем или модулем пред- и постобработки данных из других модулей. Данный модуль необходим для создания, вычисления и изменения значений переменных. Модуль позволяет производить вычисления и обработку информации с помощью языка программирования JavaScript.

Интерфейс модуля

Окно модуля очень простое и состоит всего лишь из поля для ввода и кнопок «Сохранить» и «Отмена».

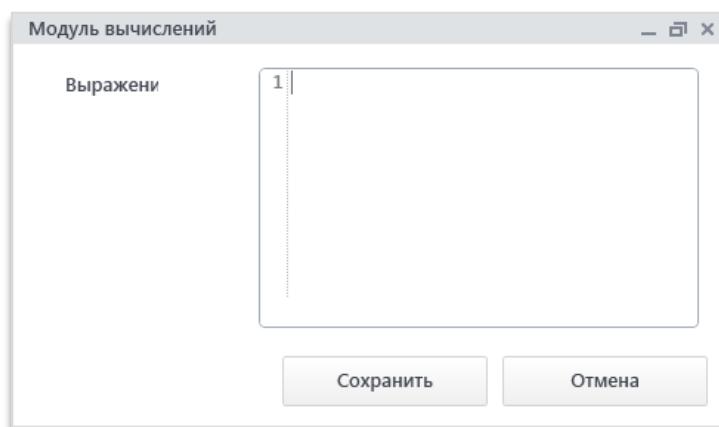


Рис. 11.31.1. Окно модуля

Поле «Выражение». В поле вводится код на языке программирования JavaScript. Возможно использование локальных переменных модуля (с объявлением их через let, var или const, но они не будут видны в других модулях скрипта робота), либо переменных студии, которые должны начинаться с символов «v.».

Модуль полностью подчиняется правилам языка JS, то есть поддерживаются все стандартные функции JS, разделение строк производится посредством ввода символа «;» или переходом строки (клавиша Enter).

Пример работы с модулем

Предположим, необходимо получить последний день предыдущего месяца. Сделаем это с помощью ввода следующего кода:

```

let date = new Date();
let year = date.getFullYear();
let month = date.getMonth();
v.lastDay = new Date(year,month,0);
v.lastDay = v.lastDay.getDate();

```

Листинг 11.31.1. Получение последнего дня предыдущего месяца

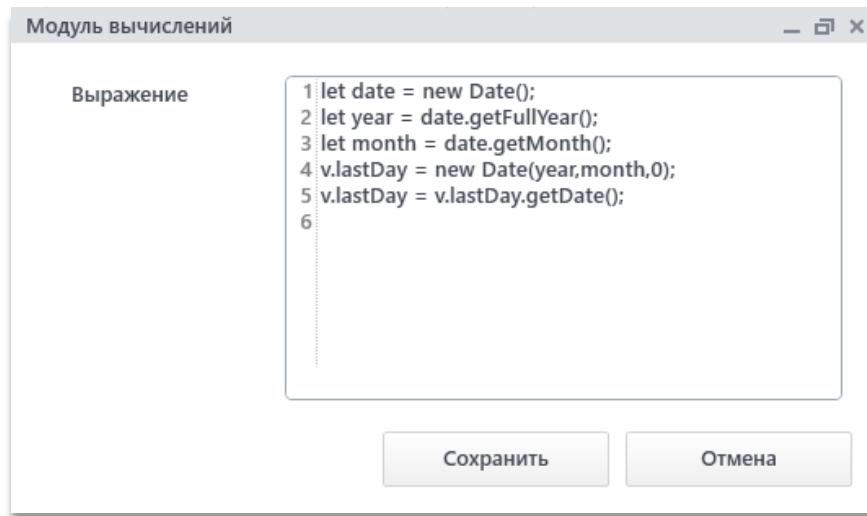


Рис. 11.31.2. Пример заполнения модуля

Что делает написанный выше код? А вот что – получаем сегодняшний день, например, сегодня 25 марта, из этой даты мы берем информацию о текущем году и месяце, затем создаем новый экземпляр даты, устанавливая в качестве года и месяца текущие, а в качестве дня – нулевой, в следствие чего язык JS создаст дату с последним днем предыдущего месяца. Более подробно:

1 строка – создание нового экземпляра даты без передачи параметров, в следствии чего создастся сегодняшняя дата (date = 25.03.2020);

2 строка – использование метода `getFullYear`, который возвращает полный год даты, к которой он применен (year = 2020);

3 строка – использование метода `getMonth`, который возвращает месяц даты, к которой он применен (month = 2, так как месяца в JS считаются с 0);

4 строка – создание нового экземпляра даты, в качестве параметров подаются текущий год, текущий месяц и 0 в качестве дня (v.lastDay = 29.02.2020);

5 строка – получение числа последнего дня предыдущего месяца (v.lastDay = 29).

В качестве знакомства с языком программирования JavaScript рекомендуем к изучению интернет-ресурс «Современный учебник JavaScript», доступный по ссылке <https://learn.javascript.ru/>.

Глава 32. Модуль «Запустить процесс»

Модуль «Запустить процесс» предназначен для запуска каких-либо приложений. Это может быть браузер, MS Word, 1С и другие.

Интерфейс модуля

Окно модуля состоит из полей «Путь к файлу», «Параметры», флажок «От имени администратора», «Пауза», «Переменная с именем процесса» и «Переменная с id процесса», а также кнопок «Сохранить» и «Отмена».

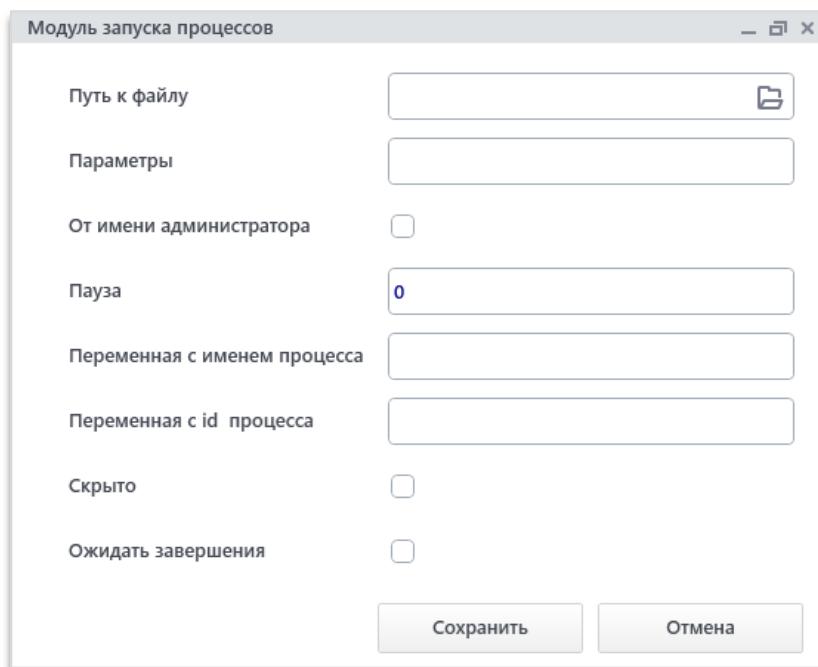


Рис. 11.32.1. Окно модуля

Поле «Путь к файлу». В данное поле вводится название стандартного процесса Windows (например, notepad – блокнот), либо путь до корневого расположения программы, например, стандартный путь до браузера Google Chrome - C:\Program Files (x86)\Google\Chrome\Application\chrome.exe.

Поле «Параметры» является необязательным полем, в него вводятся аргументы вызываемого приложения. Например, для открытия нового блокнота или чистого листа Excel-аргументы не нужны, но если необходимо открыть конкретный файл, то в аргументы передается название файла, а если открывается браузер, то аргументом может быть ссылка на сайт.

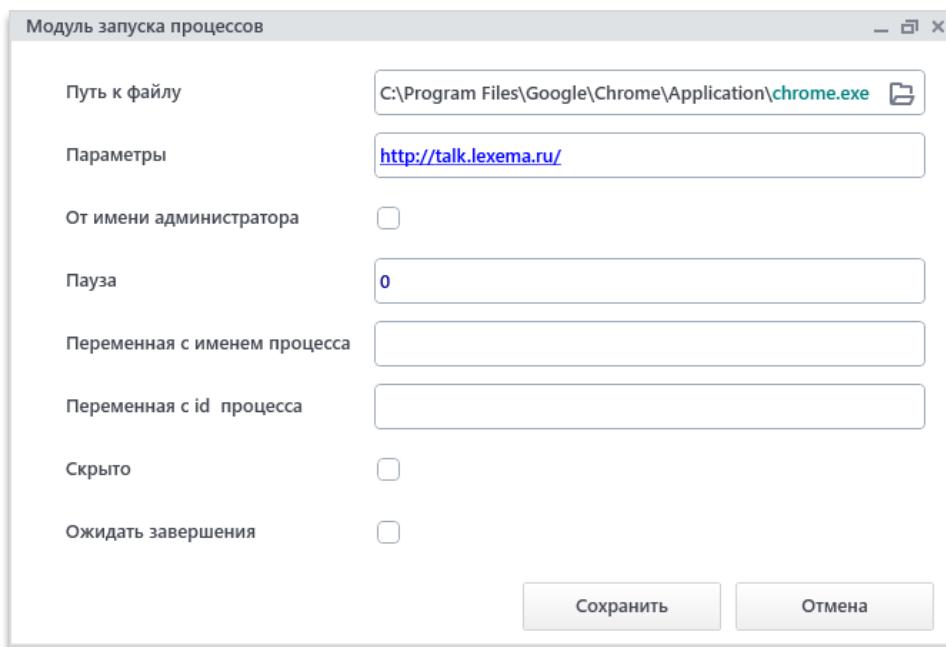


Рис. 11.32.2. Пример заполнения

Флажок «От имени администратора» запускает процесс с правами, позволяющими запускаемому процессу вносить изменения в системные файлы компьютера. *Если вы не доверяете приложению, то не стоит ставить этот флажок.*

Поле «Пауза» устанавливает время ожидания после начала запуска процесса.

Поле «Переменная с именем/id процесса». В данное поле вводятся переменные, в которые будут помещены имя и идентификатор (id) запускаемого процесса соответственно. Данные параметры нужны, чтобы в дальнейшем можно было завершить (закрыть) указанные процессы. *Если у вас задача только запустить процессы, то можете оставить эти поля пустыми.*

Флажок «Скрыто» запускает процесс со свойством hide (скрытый), то есть запускает фоновый процесс.

Флажок «Ожидать завершения» позволяет данному модулю дождаться окончания работы процесса и только затем продолжить работу робота.

Глава 33. Модуль «Завершить процесс»

Модуль «Завершить процесс» предназначен для завершения ранее запущенных процессов по их имени и идентификатору.

Интерфейс модуля

Окно состоит из двух полей – имени и идентификатора (id) завершаемого процесса.

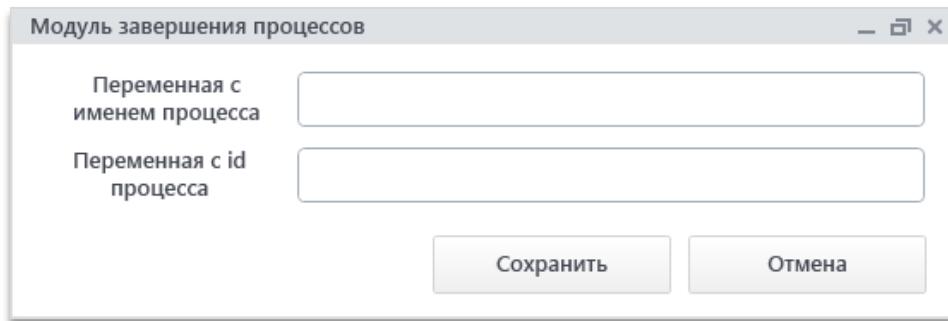


Рис. 11.33.1. Окно модуля

Если завершаемый процесс запускался через модуль «Запустить процесс», то в эти поля передаются переменные, указанные в одноименных полях окна запуска процессов. В противном случае, если известно имя процесса, который необходимо завершить, достаточно ввести только его, но (!) в таком случае *завершатся все процессы с подобным именем*. Например, если вы захотите закрыть excel-файл и напишете просто «Excel», то закроются все excel-файлы, которые были открыты.

Глава 34. Модуль «SQL»

Модуль «SQL» обеспечивает работу с базами данных. Поддерживаемые СУБД: MS SQL и PostgreSQL.

Интерфейс модуля

Данный модуль обеспечивает работу с базами данных. Поддерживаются такие СУБД, как MS SQL и PostgreSQL. Для работы с БД **необходимо заполнить все поля** окна «Модуль SQL».

Окно состоит из следующих полей: «СУБД», «Источник данных», «База данных», «Логин», «Пароль», «Время ожидания», «SQL-скрипт» и «Переменная».

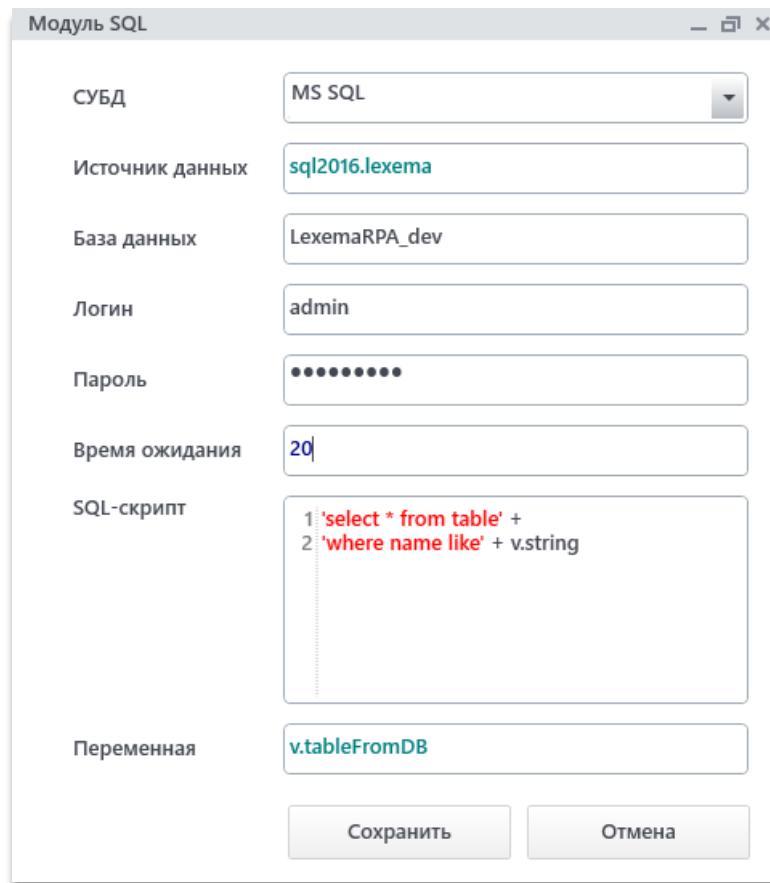


Рис. 11.34.1. Пример заполнения модуля

Поля «СУБД», «Источник данных», «База данных», «Логин» и «Пароль» обеспечивают подключение к БД.

Поле «Время ожидания». В данное поле вводится значение времени в секундах, которое должно хватить запросу на выполнение. Без этого поля робот может работать бесконечно долго из-за исключительных ситуаций, поэтому заполнение этого поля является обязательным. Для небольших запросов значение может быть равно 20 секунд, для больших – минута и более.

Поле «SQL-скрипт». В данное поле вводится скрипт на языке SQL, поддерживается ввод переменных студии. При использовании переменных студии необходимо подать в это поле строку в следующем виде: весь скрипт должен быть взят в кавычки, за исключением переменных студии (как показано на Рис. 11.34.1).

Поле «Переменная». В поле вводится название переменной, в которую будет записан ответ от базы данных в виде таблицы. *В одном SQL-модуле может возвращаться только одна таблица, последующие будут проигнорированы.*

Глава 35. Модуль «HTTP OData»

Http-запросы – запросы для передачи информации между сайтами (или серверами) и клиентами (например, нами).

Модуль «HTTP OData» предназначен для отправки запросов по протоколу *http*⁹ к интеграционному сервису 1С – OData.

Интерфейс модуля

Данный модуль состоит из полей: «Сылка», «Логин», «Пароль», «Переменная» и «Тип запроса».

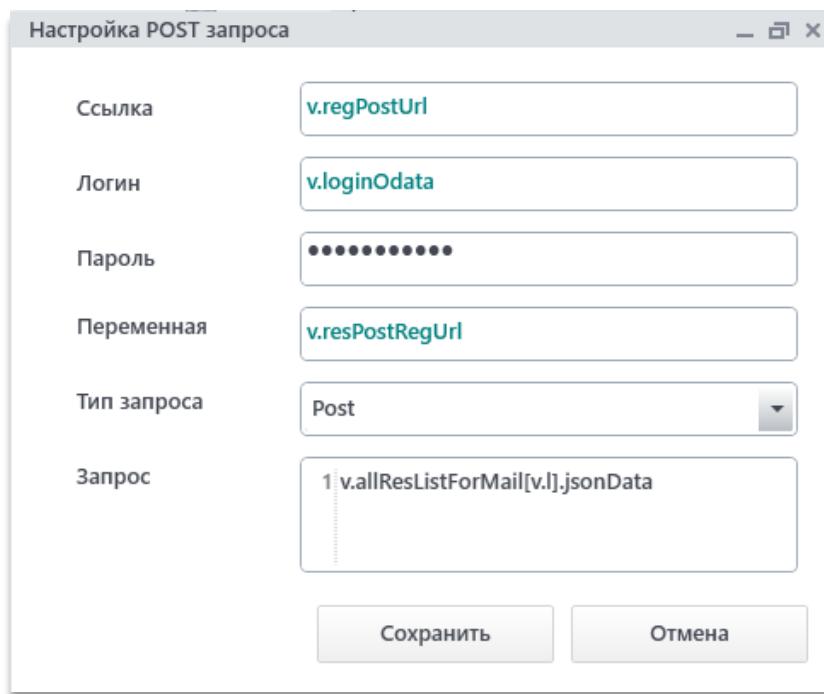


Рис. 11.35.1. Пример настройки GET запроса

Поле «Сылка». В поле вводится необходимый URL-адрес.

Поле «Переменная». В поле вводится название переменной, в которую будет помещен ответ.

Поля «Логин» и «Пароль». Предназначены для ввода логина и пароля при авторизации.

Поле «Тип запроса». Позволяет выбрать необходимый тип запроса. Доступны методы «GET», «PUT», «POST» и «PUTCH».

⁹ HTTP (англ. HyperText Transfer Protocol — «протокол передачи гипертекста») — протокол прикладного уровня передачи данных изначально — в виде гипертекстовых документов в формате «HTML», в настоящий момент используется для передачи произвольных данных.

Поле «Запрос». В поле вводится HTTP-сообщение, согласно общим правилам его структуры. Доступно к заполнению только при выборе типов запроса «PUT», «POST» и «PUTCH».

Глава 36. Модуль «HTTP запрос»

Http-запросы – запросы для передачи информации между сайтами (или серверами) и клиентами (например, нами).

Модуль «HTTP запрос» предназначен для отправки запросов по протоколу *http*.

Интерфейс модуля

Данный модуль состоит из полей: «Ссылка», «Переменная», «Тип запроса» и вкладок Headers и Body с одинаковыми полями Name и Value. Внизу модуля расположена таблица с добавленными значениями аргументов.

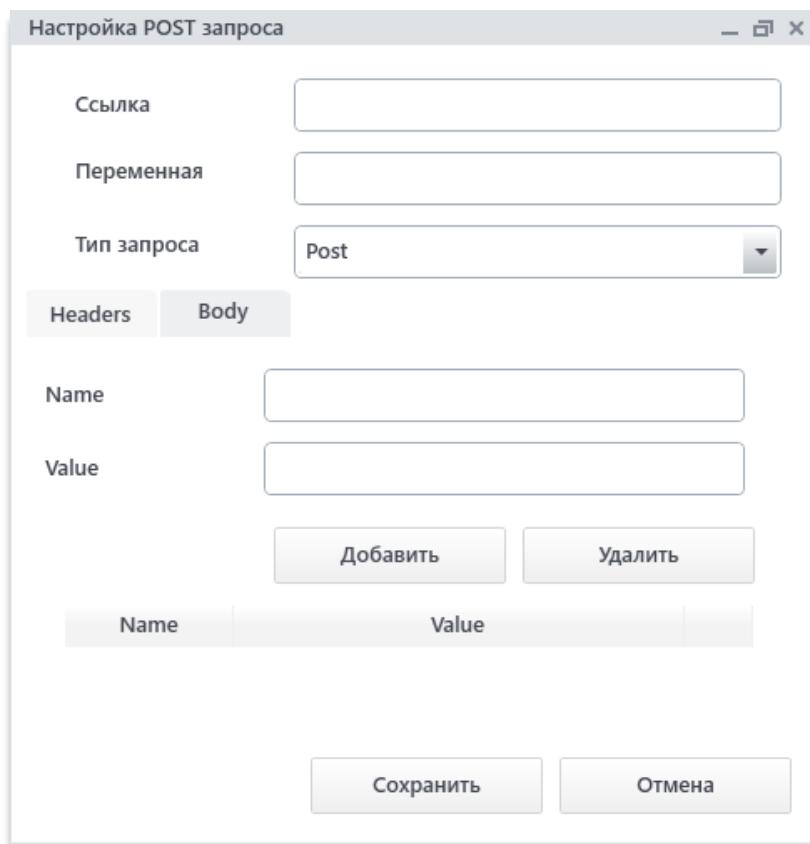


Рис. 11.36.1. Интерфейс модуля «HTTP запрос»

Поле «Ссылка». В поле вводится необходимый URL-адрес.

Поле «Переменная». В поле вводится название переменной, в которую будет помещен ответ.

Поле «Тип запроса». Позволяет выбрать необходимый тип запроса. Доступны методы «GET» и «POST».

Поля «Name» и «Value» вкладок «Headers» и «Body» предназначены для составления http-запроса.

Глава 37. Модуль «FTP запрос».

Модуль позволяет отправлять запросы по **FTP**¹⁰ протоколу.

Интерфейс модуля

Модуль состоит из нескольких полей для подключения к серверу, ввода названия переменной и выпадающего списка с выбором действия.

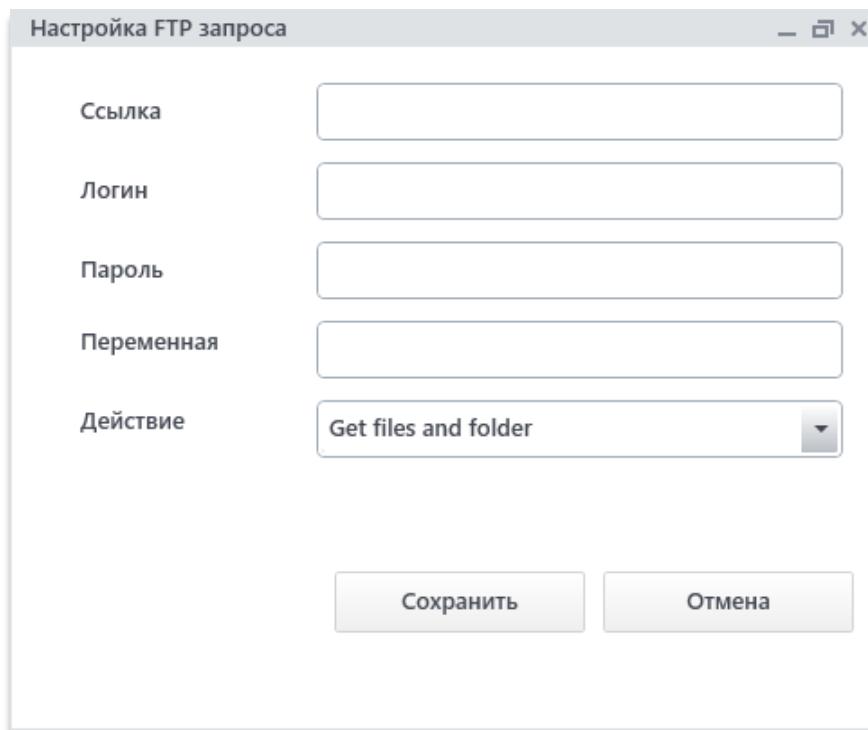


Рис. 11.37.1. Интерфейс модуля «FTP запрос»

Поле «Ссылка» предназначено для ввода ссылки для подключения к FTP серверу.

Поле «Логин» заполняется логином подключающегося к серверу пользователя.

Поле «Пароль» заполняется паролем пользователя для доступа к серверу.

Поле «Переменная» предназначено для ввода названия переменной, в которую будет помещен результат выполнения модуля.

Пол «Действие» предоставляет перечень доступных действий с FTP сервером.

¹⁰ FTP (англ. File Transfer Protocol) — протокол передачи файлов по сети, построенный на архитектуре «клиент-сервер». Широко используется для распространения ПО и доступа к удалённым хостам.

Глава 38. Модуль «JSON»

Данный модуль предназначен для обработки данных в формате JSON – парсинга JSON-объектов.

Интерфейс модуля

В окне модуля расположены поля для ввода значений: «Переменная» и «JSON», и кнопки «Сохранить» и «Отмена».

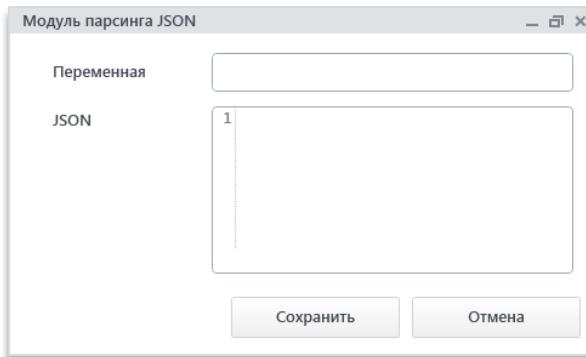


Рис. 11.38.1. Окно модуля

Поле «Переменная». В поле «Переменная» вводится название переменной, которая будет преобразована в объектную переменную с полями, имеющимися в JSON-объекте.

Поле «JSON». В поле необходимо описать объект в формате JSON, в виде пар «ключ»: «значение», обрамлённых в фигурные скобки. JSON-объект может быть описан непосредственно текстом в поле «JSON» либо быть созданным в процессе выполнения других модулей. В случае формирования JSON-объекта в других модулях, в поле необходимо ввести название переменной, в которой лежит необходимый объект.

ПРИМЕР.

Пусть у нас есть текст в формате JSON. Введем его в поле JSON модуля, а результат пусть запишется в переменную v.a.

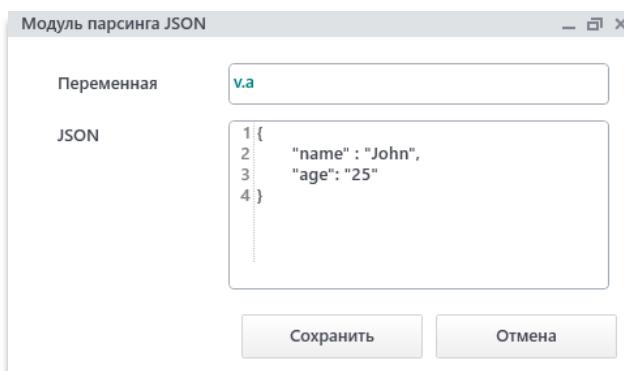


Рис. 11.38.2. Пример заполнения

В результате переменная `v.a` станет объектом с полями «`name`» и «`age`». К полям объекта можно обратиться, написав название поля через точку после переменной. Проверить значение переменной можно с помощью функционала «Узнать значение переменной», доступного по кнопке  в обозревателе решений.

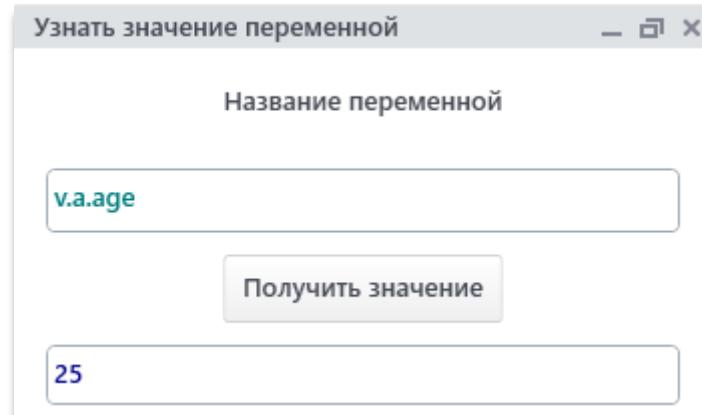


Рис. 11.38.3. Получение значения из поля «`name`» переменной «`a`»

На Рис. 11.38.3 видно, что на основе текста формата JSON создалась переменная с указанными в нем полями и заполненными значениями.

Глава 39. Модуль «XML»

Модуль «XML» предназначен для парсинга текста формата XML. Результатом работы модуля будет объект с полями и свойствами, указанными в исходном тексте.

Интерфейс модуля

Окно модуля состоит из двух полей – поля «Переменная» и «XML».

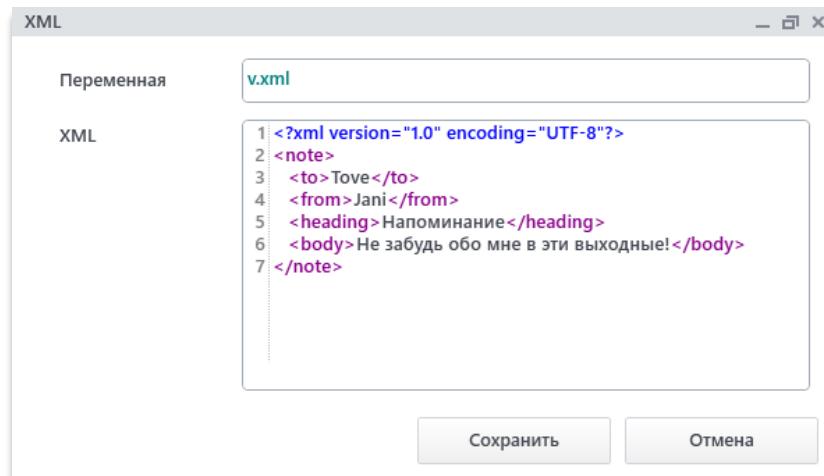


Рисунок 11.XI.1. Окно модуля «XML»

Поле «Переменная» заполняется названием переменной, в которую будет возвращен объект – результат работы модуля.

Поле «XML» заполняется текстом в формате XML или переменной, содержащей этот текст.

Результат работы модуля при его настройке, указанной на Рисунок 11.XI.1.Окно модуля «XML».:

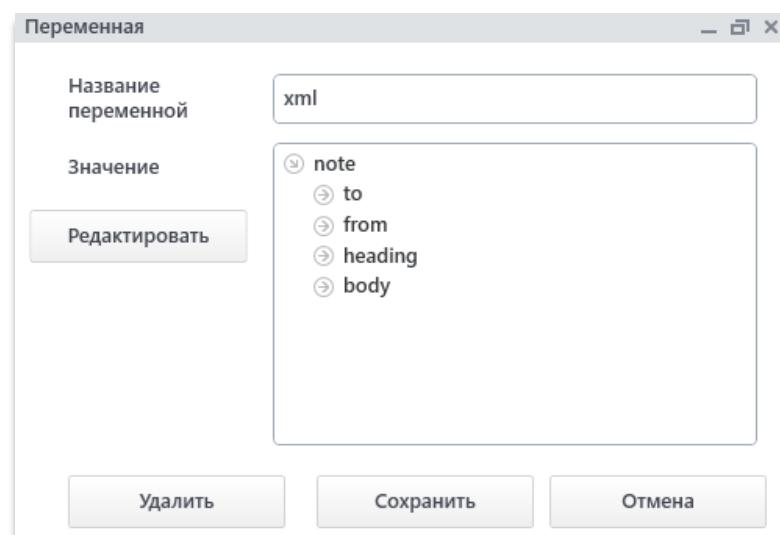


Рисунок 11.XI.2. Объект – результат парсинга XML

Глава 40. Модуль «PDF»

Данный модуль позволяет работать с PDF файлами. В его возможности входит получение текста из документа, изображений, и объединение нескольких PDF файлов.

Интерфейс модуля

Модуль состоит из 4-х полей, доступность которых определяется выбранным действием, кнопок управления создаваемым действием и таблицей-списком созданных действий.

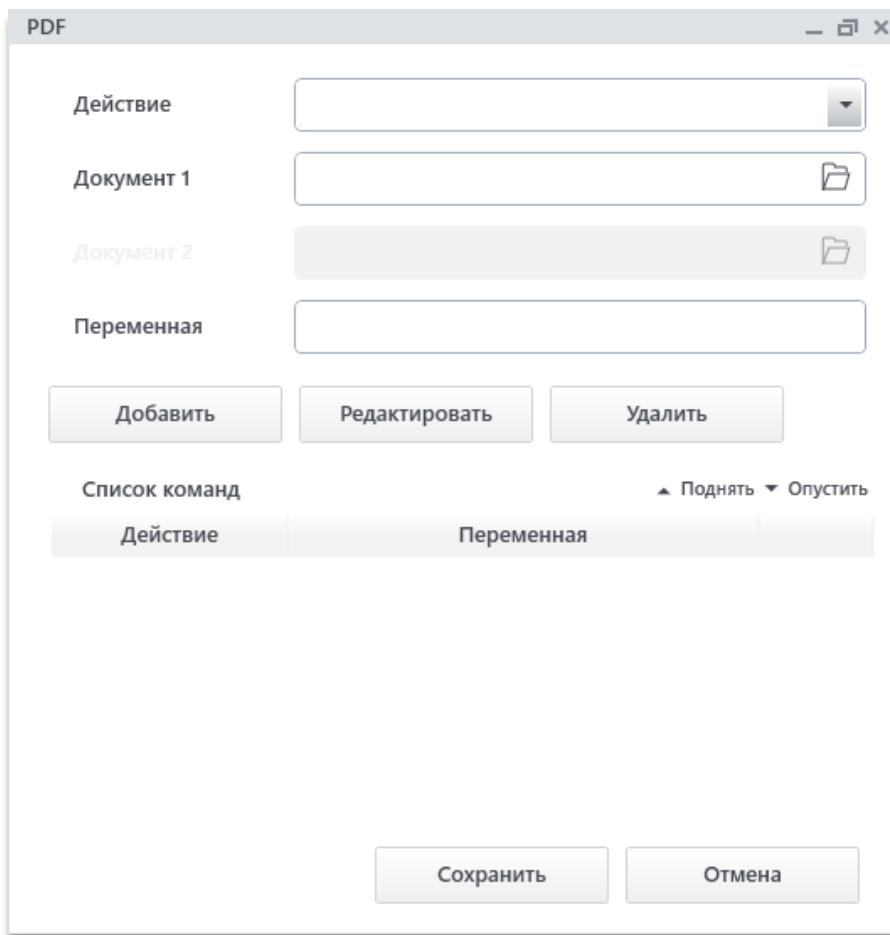


Рис. 11.40.1. Интерфейс модуля «PDF»

Поле «Действие». Представляет собой выпадающий список из доступных в модуле действий.

Поле «Документ 1». Поле, в которое вводится путь до рабочего файла. По кнопке в поле открывается диалог выбора файла.

Поле «Документ 2». Поле, в которое вводится путь до второго рабочего файла, используется для действий, требующих несколько файлов (например, объединение 2-х PDF). По кнопке в поле открывается диалог выбора файла.

Поле «Переменная». Поле, для ввода названия переменной, в которую будет возвращен результат работы модуля.

Перечень и описание действий

Получить text

Действие позволяет получить текст из файла PDF, если тот является непосредственно текстом, а не изображением. Полученный текст будет записан в указанную переменную.

Действие	Получить text
Документ 1	C:\Users\user\Desktop\test.pdf
Папка	
Переменная	v.text

Рис. 11.40.2. Пример настройки действия «Получить text»

Получить image

Действие получает все изображения из PDF файла и сохраняет их в формате .jpg в указанную в поле «Папка» директорию. В указанную переменную будет записан список путей до выделенных изображений.

Действие	Получить image
Документ 1	C:\Users\user\Desktop\test.pdf
Папка	C:\Users\user\images
Переменная	v.text

Рис. 11.40.3. Пример настройки действия «Получить image»

Объединить документы

Действие объединяет два указанных PDF файла и сохраняет их в файле, который необходимо указать в поле «Значение».

Действие	Объединить документы
Документ 1	C:\Users\user\Desktop\test1.pdf
Документ 2	C:\Users\user\Desktop\test2.pdf
Значение	C:\Users\user\Desktop\test.pdf

Рис. 11.40.4. Пример настройки действия «Объединить документы»

Глава 41. Модуль «Пауза»

Модуль «Пауза» предназначен для создания некоторой паузы между модулями.

Вся настройка модуля заключается в заполнении поля «Пауза» - количество секунд, которые необходимо подождать. Допускается использование переменной.

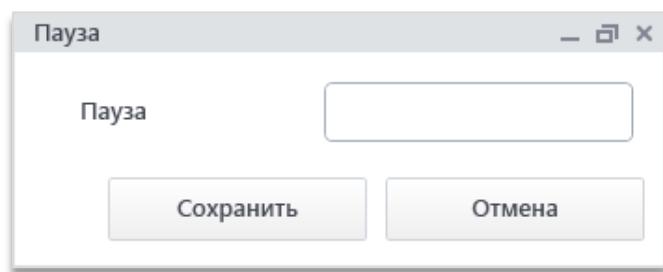


Рис. 11.41.1. Окно модуля «Пауза»

Глава 42. Модуль «Прокси»

Модуль «Прокси» предназначен для создания подключения к прокси-серверу. Для использования модуля необходимо заполнить адрес прокси-сервера.

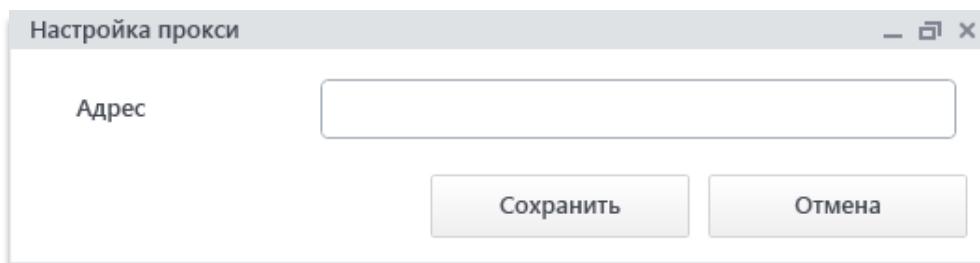


Рис. 11.42.1. Окно модуля «Прокси»

Глава 43. Модуль «Создать скриншот»

Данный модуль позволяет делать скриншот экрана во время работы робота, что удобно для проведения отладки кликерных роботов (роботов, преимущественно работающих на модуле Кликер).

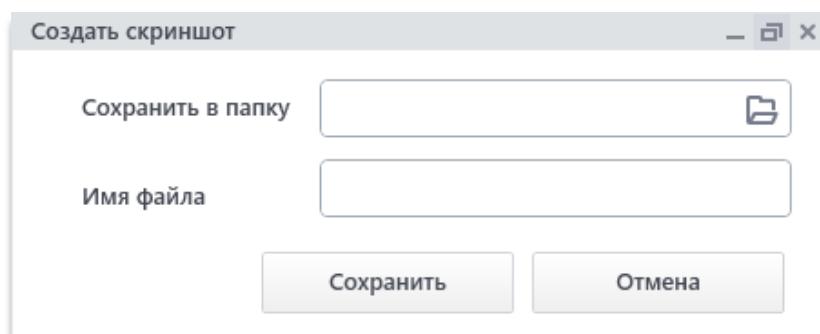


Рис. 11.43.1. Окно модуля «Создать скриншот»

Глава 44. Модуль «Лог»

Модуль «Лог» предназначен для вывода некоторых сообщений или значений переменных в «Отладочный вывод» во время работы робота.

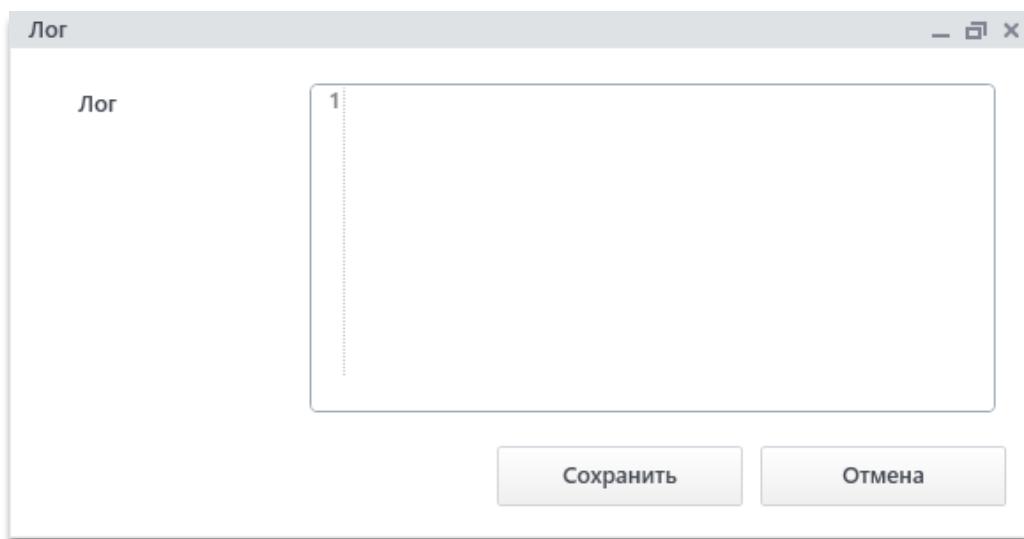


Рис. 11.44.1. Окно модуля «Лог»

ЧАСТЬ XII. ОРКЕСТРАТОР

Глава 45. Общая информация

Оркестратор роботов - сервис запуска роботов по составленному расписанию и/или некоторым событиям. Доступ к оркестратору осуществляется по нажатию кнопки  в панели управления главного окна студии. После нажатия на кнопку будет открыто окно, позволяющее опубликовать нового робота и/или настроить расписание запусков уже опубликованных роботов.

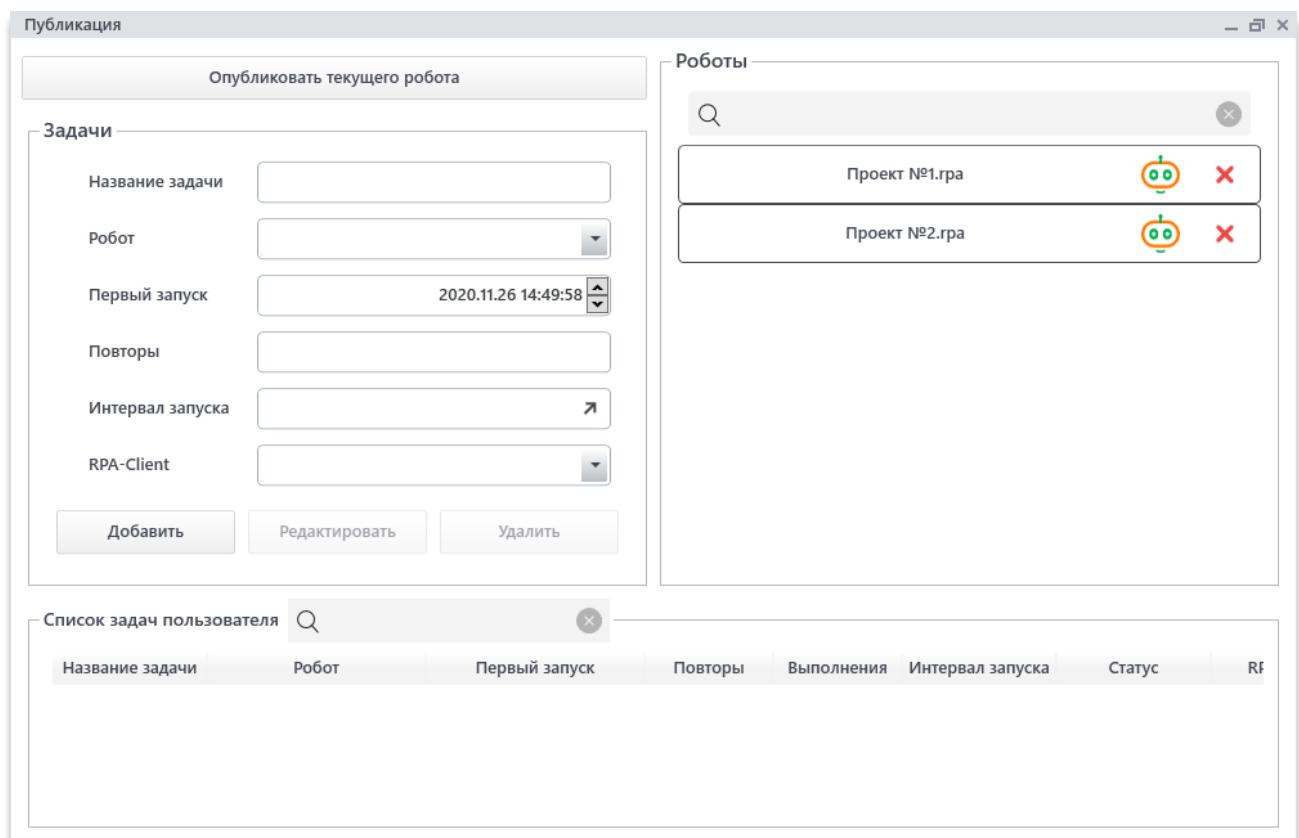


Рис. 12.45.1. Окно оркестратора

Оркестратор состоит из 3-х разделов и кнопки публикации текущего робота. Разделы оркестратора:

- Задачи – раздел, позволяющий создавать новую задачу;
- Роботы – список всех опубликованных текущим пользователем роботов;
- Список задач пользователя – список всех задач, созданных текущим пользователем.

Кнопка «Опубликовать текущего робота» добавляет активного на данный момент в студии робота в список опубликованных роботов. Если данный робот был уже однажды загружен в оркестратор, то повторное нажатие кнопки его обновит. Привязка робота идет не

по имени, а по его идентификатору, который присваивается роботу при его первоначальном создании.

Создание задач описано далее.

После создания всех задач необходимо просто закрыть окно оркестратора.

Для работы оркестратора необходимо установить программу Lexema-RPA Client, которая, находясь в фоновом режиме работы на компьютере, просматривает созданные вами задачи и запускает роботов в соответствии с их расписанием. Для её скачивания необходимо посетить сайт <http://lexema.ru/solutions/lexema-rpa-programmnye-roboty/lexema-rpa-studio/> и выбрать там «Скачать Lexema-RPA Client». Инструкция по использованию находится там же.

Глава 46. Создание новой задачи

Для создания расписания запуска одного робота нужно создать задачу. Для настройки задачи необходимо заполнить 6 полей – это наименование задачи, выбор робота из опубликованных, время первого запуска робота, количество повторов и интервал запуска, машина для запуска, на которой запущен RPA Client.

Название задачи	<input type="text"/>
Робот	<input type="button" value="▼"/>
Первый запуск	2020.11.26 14:49:58 <input type="button" value="▲"/> <input type="button" value="▼"/>
Повторы	<input type="text"/>
Интервал запуска	<input type="button" value="↗"/>
RPA-Client	<input type="button" value="▼"/>

Рис. 12.46.1. Перечень полей для настройки задачи запуска

Наименование задачи. Название задачи для запуска конкретного робота, идентификатор, по которому вам будет удобнее ориентироваться среди всех ваших задач.

Робот. Выбор робота, для которого составляется задача запуска. Поле представлено выпадающим списком, в котором предложены все ваши опубликованные роботы.



Рис. 12.46.2. Выпадающий список с перечнем опубликованных роботов

Первый запуск. Время первого запуска робота, после которого он будет запускаться с определенным интервалом. Выбираются и дата, и время.

Повторы. Общее количество повторных запусков робота. Если робота повторять не нужно, и он должен отработать только 1 раз, количество повторов равно 0. Если необходимо, чтобы робот отрабатывал по 2 раза в день в течение недели, не учитывая его первый запуск, то количество повторов равно $2*7-1 = 13$.

Интервал запуска. Время, через которое необходим повторный запуск робота (если повторы есть). Для настройки поля необходимо нажать на кнопку-стрелочку в углу поля.



Рис. 12.46.3. Кнопка открытия настроек интервала

По нажатию на неё откроется вспомогательное окно выбора интервала запуска робота.

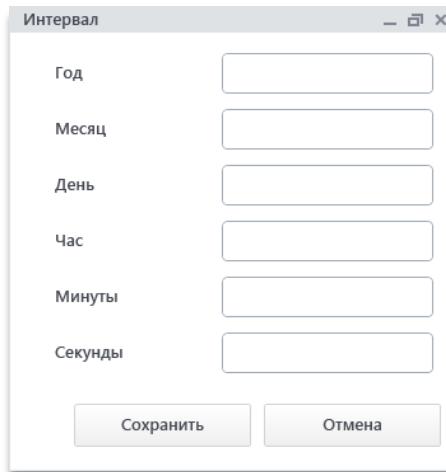


Рис. 12.46.4. Настройка интервала запуска робота

Если необходимо, чтобы робот запускался 25 числа каждого месяца, необходимо в первом запуске установить дату ближайшего 25 числа месяца, а настройке интервала ввести значение «1» в строке «Месяц», то есть, осуществлять повторы каждый месяц.

Междуд каждым повтором будет установлен одинаковый интервал. Если необходимо запускать робота с *разными* временными интервалами, то необходимо создать несколько задач его запуска, например, если требуется запуск в 10:00 и в 13:00 одного дня на протяжении недели, то необходимо создать *две* задачи, количество повторений которых равно 6 (каждый день, не считая первого запуска), интервал запуска – 24 часа или 1 день, отличаться у этих задач будут только время первого запуска – у одной в 10:00 дня, когда нужно начать, у другой – в 13:00.

После заполнения нужных строк в окне интервала и нажатию на кнопку «Сохранить», в поле «Интервал» отобразится строка вида «год:месяц:день:час:минуты:секунды», например, при установке интервала в 12 часов, строка будет иметь вид «0:0:0:12:0:0». Поле «Интервал» является **обязательным**, если количество повторов отличное от 0.

RPA-Client. Если существует несколько разных машин (компьютеров), на которых авторизован один и тот же пользователь и запущены программы Lexema RPA-Client, то данное поле позволяет выбрать, на какой из этих машин необходимо запускать выбранного робота. Если неважно, на какой конкретно машине должен запускаться робот, то поле можно оставить пустым, тогда робот будет запускаться на одной из свободных машин.

Пример заполненной задачи:

Название задачи	Задача №1
Робот	Проект №1.gra
Первый запуск	2020.11.26 12:00:00
Повторы	5
Интервал запуска	0:0:0:3:30:0
RPA-Client	

Рис. 12.46.5. Пример настройки задачи

По нажатию кнопки «Добавить», расположенной под полями для настройки задач, задача будет добавлена в список задач.

Список задач пользователя					
Название задачи	Робот	Первый запуск	Повторы	Выполнения	Интервал запуска
Задача №1	Проект №1.gra	26.11.2020 12:00:00	5	0	0:0:0:3:30:0

Рис. 12.46.6. Список задач пользователя с добавленной задачей

Для редактирования существующей задачи необходимо найти её в списке задач и нажать на неё (1 на Рис. 12.46.7). После этого будут заполнены поля для настройки задачи соответствующими полями. Затем, после редактирования полей (2 на Рис. 12.46.7), необходимо нажать на кнопку «Редактировать» и изменения сохранятся (3 на Рис. 12.46.7).

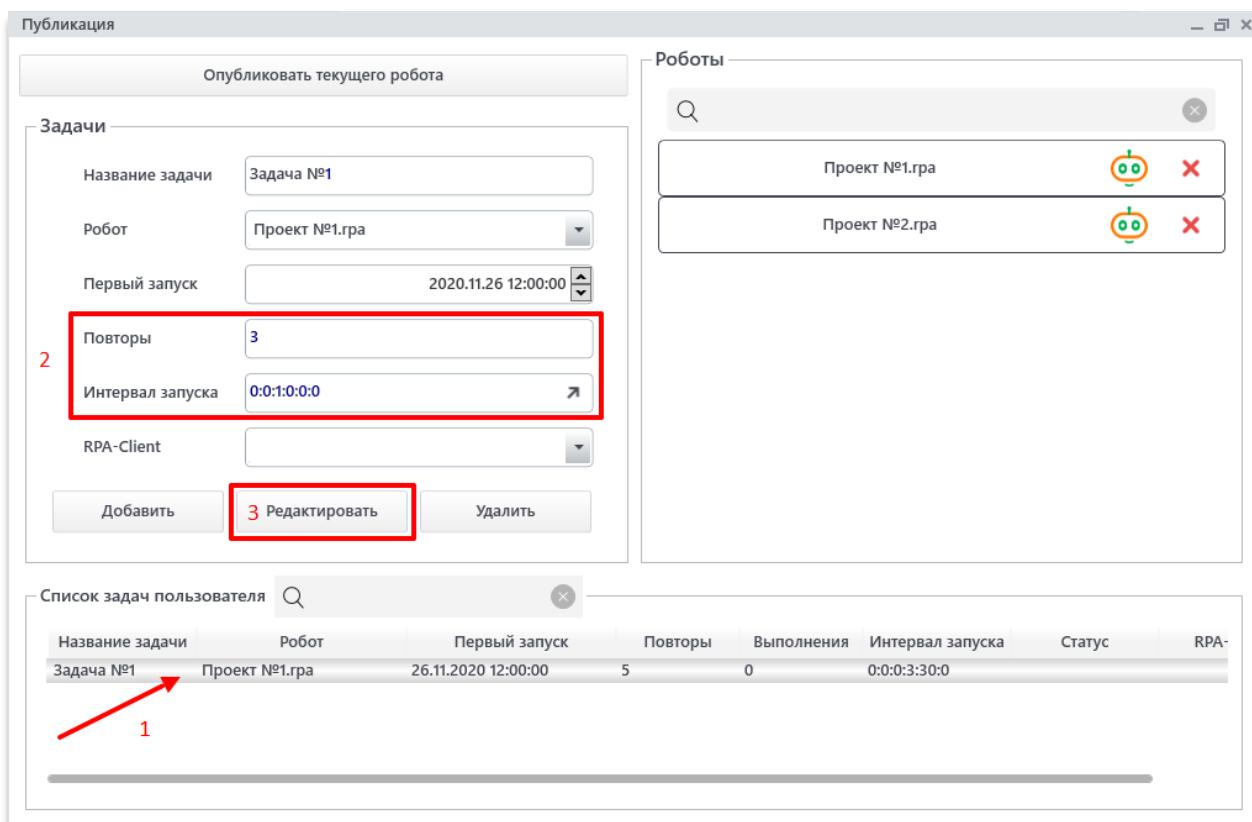


Рис. 12.46.7. Пример редактирования задачи

Для удаления задачи её нужно найти и выбрать в списке задач пользователя и затем нажать кнопку «Удалить».

Глава 47. Раздел «Роботы»

Данный раздел состоит из поля поиска робота по его названию и списка опубликованных роботов. При публикации, роботу присваивается уникальный идентификатор, поэтому названия роботов в списке опубликованных могут повторяться.



Рис. 12.47.1. Раздел «Роботы»

Для загрузки, отладки или редактирования существующего робота, необходимо открыть его как проект в студии. Для этого необходимо нажать на иконку Lexema-RPA соответствующего робота.

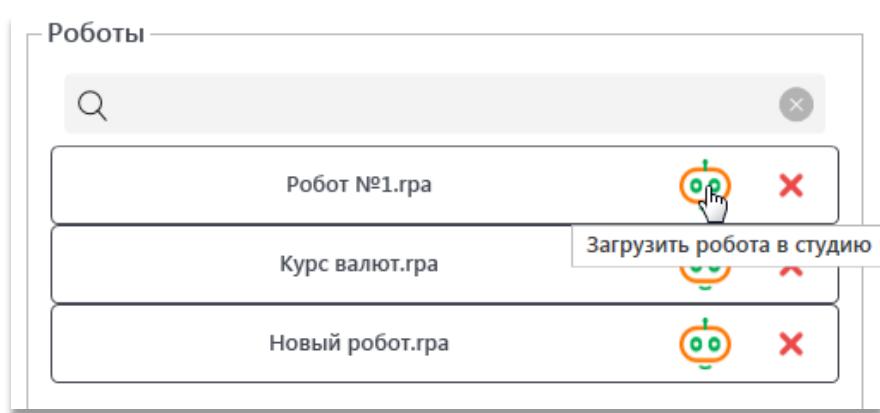


Рис. 12.47.2. Кнопка «Загрузить робота в студию»

После редактирования робота необходимо нажать на кнопку «Опубликовать текущего робота», после чего редактируемый робот сохранит свои изменения.

Для удаления робота необходимо нажать на красный крестик в его блоке.

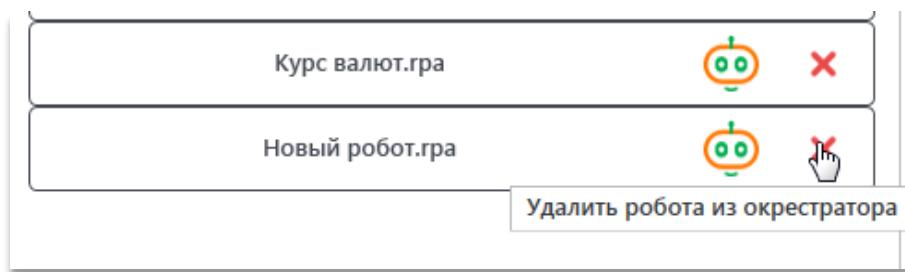


Рис. 12.47.3. Кнопка удаления робота из оркестратора

Глава 48. Просмотр истории выполнений

При нажатии правой кнопкой мыши по задаче в списке задач появляется контекстное меню, позволяющее перейти к статистике выполнений или логам задачи.

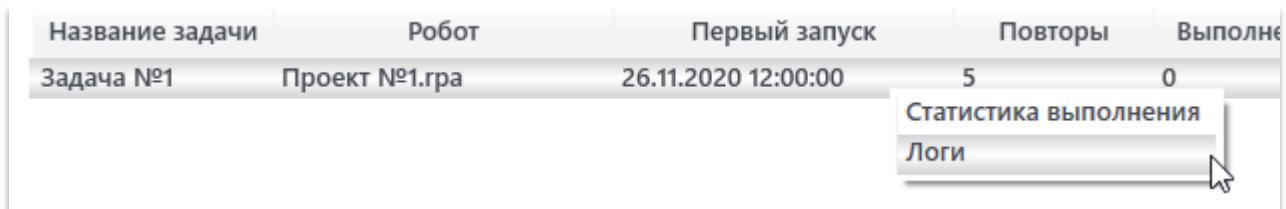


Рис. 12.48.1. Контекстное меню задачи

Статистика выполнения открывается так же при двойном клике по задаче.

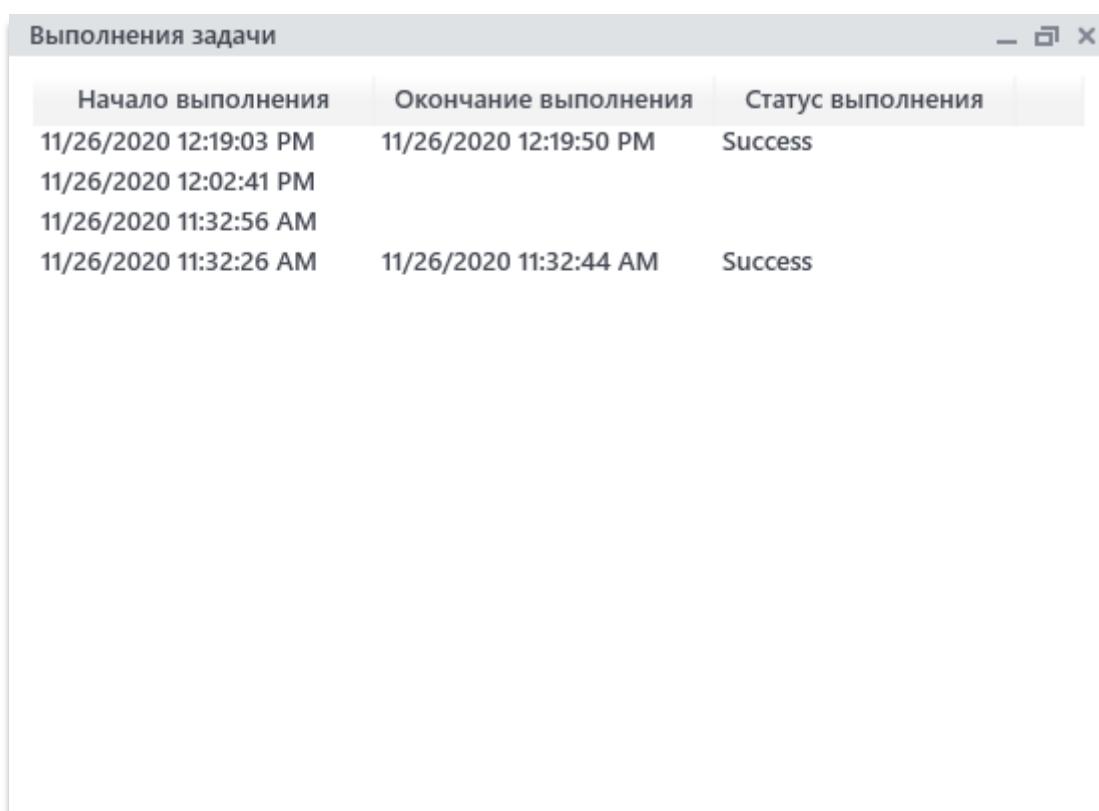


Рис. 12.48.2. Статистика выполнения

В данном окне отображаются состояния задачи – время запуска, время окончания и статус работы – Success (успешно) или Failed (провалено). Если строка пустая, значит робот по каким-то причинам был приостановлен, например был остановлен клиент или убит процесс в «Диспетчере задач».

В окне логов отображается более детальная информация – начало и окончание работы модуля робота.

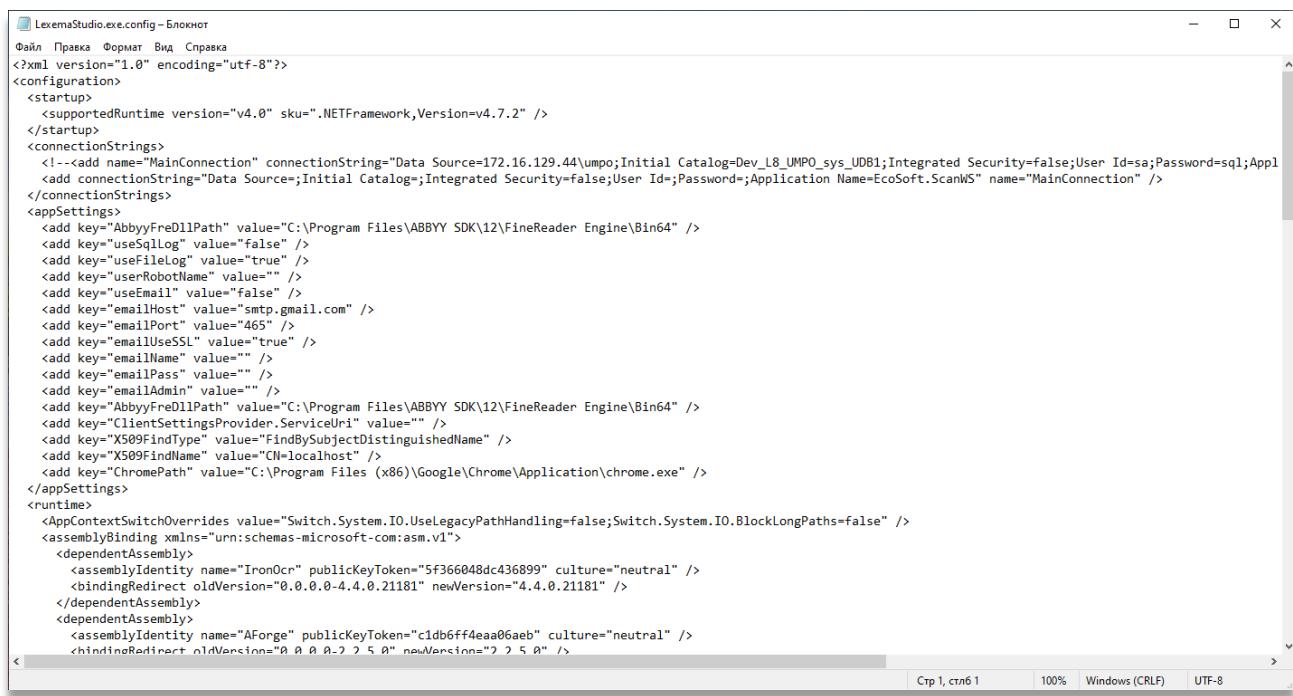
Время	Статус	Описание
11/26/2020 10:45:02 AM	True	Module TryModule(Примечание) end
11/26/2020 10:45:02 AM	True	Module SqlModule(Планирование) end
11/26/2020 10:45:01 AM	true	Module SqlModule(Планирование) start
11/26/2020 10:45:01 AM	true	Module TryModule(Примечание) start
11/25/2020 9:28:51 AM	True	Module TryModule(Примечание) end
11/25/2020 9:28:51 AM	True	Module SqlModule(Планирование) end
11/25/2020 9:28:50 AM	true	Module SqlModule(Планирование) start
11/25/2020 9:28:50 AM	true	Module TryModule(Примечание) start
11/24/2020 10:50:11 AM	true	Module TryModule(Примечание) start
11/23/2020 9:27:52 AM	True	Module TryModule(Примечание) end
11/23/2020 9:27:52 AM	True	Module SqlModule(Планирование) end
11/23/2020 9:27:49 AM	true	Module SqlModule(Планирование) start
11/23/2020 9:27:49 AM	true	Module TryModule(Примечание) start

Рис. 12.48.3. Логи

ЧАСТЬ XIII. КОНФИГУРИРОВАНИЕ LEXEMA-RPA STUDIO

Глава 49. Файл конфигурации

Lexema-RPA Studio содержит некоторые настройки, которые подходят большинству пользователей, но в некоторых случаях они должны быть изменены. Такие настройки вынесены в файл конфигурации, который находится в папке с программой и называется «Lexema-RPA Studio.exe.config». Открыть данный файл можно любым текстовым редактором, например Блокнотом.



```

LexemaStudio.exe.config - Блокнот
Файл Правка Формат Вид Справка
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
  </startup>
  <connectionStrings>
    <!--<add name="MainConnection" connectionString="Data Source=172.16.129.44\umpo;Initial Catalog=Dev_L8_UMP0_sys_UDB1;Integrated Security=false;User Id=sa;Password=sql;App=App" />
    <add connectionString="Data Source=.;Initial Catalog=.;Integrated Security=false;User Id=.;Password=.;Application Name=EcoSoft.ScanWS" name="MainConnection" />
  </connectionStrings>
  <appSettings>
    <add key="AbbyyFreDllPath" value="C:\Program Files\ABBYY SDK\12\FineReader Engine\Bin64" />
    <add key="useSqllog" value="false" />
    <add key="usefileLog" value="true" />
    <add key="userRobotName" value="" />
    <add key="useEmail" value="false" />
    <add key="emailHost" value="smtp.gmail.com" />
    <add key="emailPort" value="465" />
    <add key="emailUseSSL" value="true" />
    <add key="emailName" value="" />
    <add key="emailPass" value="" />
    <add key="emailAdmin" value="" />
    <add key="AbbyyFreDllPath" value="C:\Program Files\ABBYY SDK\12\FineReader Engine\Bin64" />
    <add key="ClientSettingsProvider.ServiceUrl" value="" />
    <add key="X509FindType" value="FindBySubjectDistinguishedName" />
    <add key="X509FindName" value="CN=localhost" />
    <add key="ChromePath" value="C:\Program Files (x86)\Google\Chrome\Application\chrome.exe" />
  </appSettings>
  <runtime>
    <AppContextSwitchOverrides value="Switch.System.IO.UseLegacyPathHandling=false;Switch.System.IO.BlockLongPaths=false" />
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="IronOcr" publicKeyToken="5f366048dc436899" culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-4.4.0.21181" newVersion="4.4.0.21181" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="AForge" publicKeyToken="c1db6ff4eaa06aeb" culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-2.2.5.0" newVersion="2.2.5.0" />
      </dependentAssembly>
    </assemblyBinding>
  </runtime>

```

Рис. 13.49.1. Файл конфигурации

Настройки, которые можно изменять, находятся в блоке `appSettings`. Приведенные ниже можно изменять пользователям, остальные не рекомендуется. Доступные для изменения настройки:

1. `AbbyyFreDllPath` - путь до папки с драйвером ABBYY (для пользователей, приобретших лицензию ABBYY);
2. `useFileLog` - использовать логирование в текстовый файл, по умолчанию значение `true`, запись ведется во временную директорию системы, в папку Lexema-RPA;
3. `useEmail` – использовать логирование с отправкой письма с ошибкой на почту. Настройки почты вводятся в поля `emailHost`, `emailPort`, `emailUseSSL`, `emailName` и `emailPass`. В поле `emailAdmin` вводится адрес, на который должно прийти письмо;
4. `ChromePath` – путь до браузера Chrome для модуля «Работа с браузером».