Создание робота по распознаванию PDF-документа

1. Базовые понятия

В данном разделе определяется ряд базовых понятий, которые будут использоваться в дальнейшем (рисунок 1.1):

- *Рабочая область* – центральное окно в программе PIX Studio после создания проекта
- *Окно активностей* – левое окно в программе PIX Studio после создания проекта
- *Активности* – базовые функции, из которых конструируется робот
- *Группа активностей* – активности, объединённые единой идей своего функционала. В дальнейшем для ясности активности будут именоваться следующим образом «Группа активностей/Название активности»
- *Использовать/применить активность* – перетащить активность из окна активностей в рабочую область. Для выполнения данного действия требуется нажать ЛКМ на активность и, зажав ЛКМ, перенести курсор в рабочую область, а затем отпустить ЛКМ

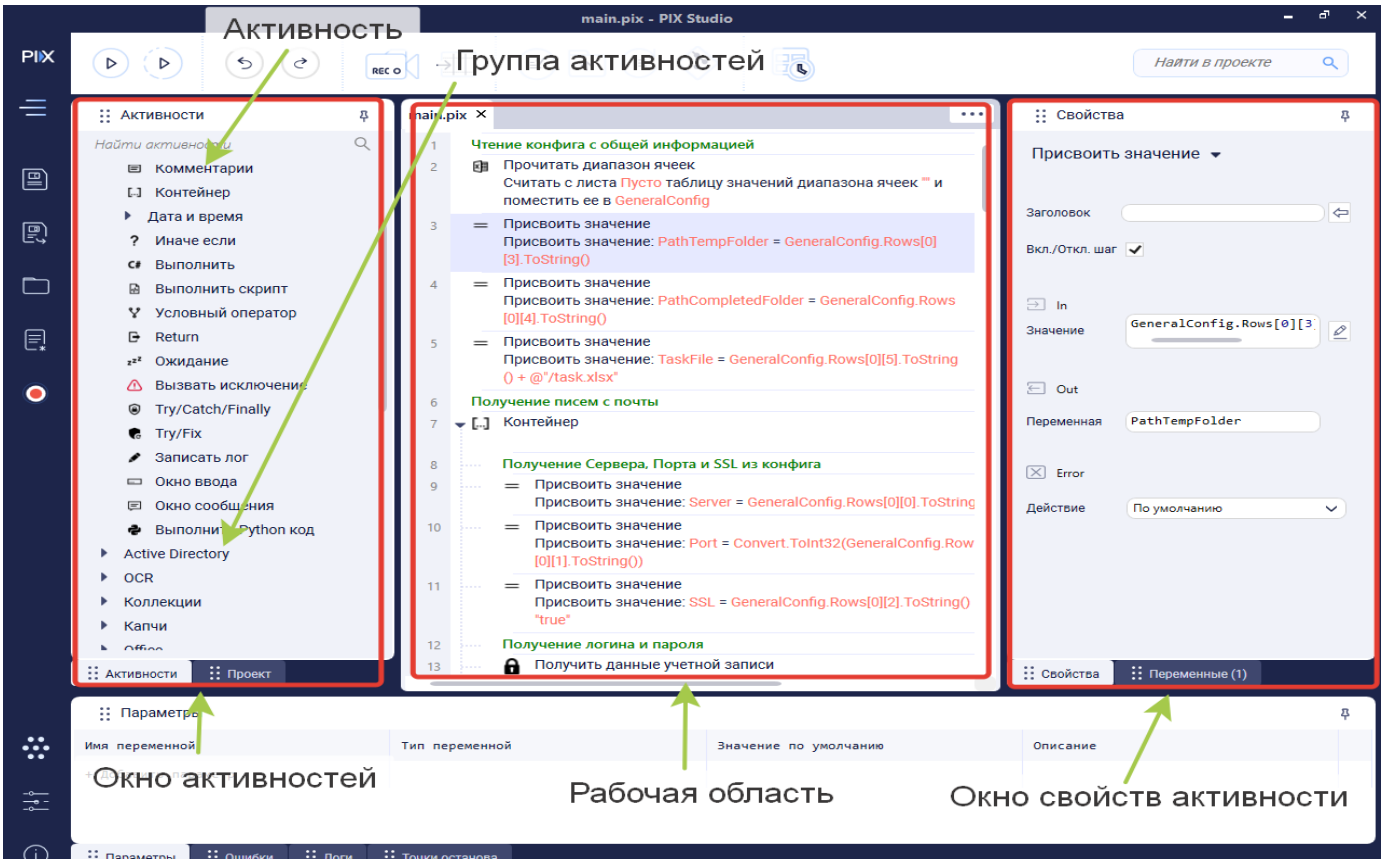


Рисунок 1.1 Окно редактора PIX Studio

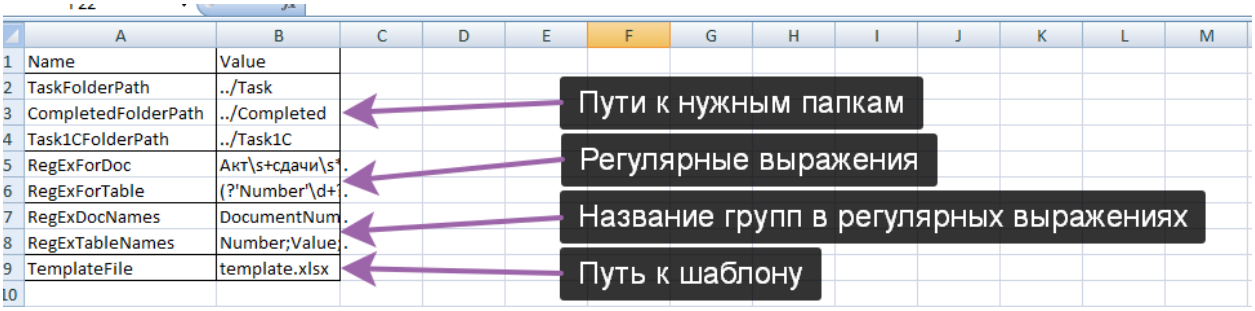
2. Постановка задачи

Робот должен проверить папку, в случае наличия PDF документов робот считывает с них текст. Затем считанные данные заносятся в excel таблицу в соответствующие поля. Данные из полученного excel документа заносятся в 1С.

3. Предварительные действия

Прежде чем приступить к созданию робота, желательно выполнить ряд подготовительных действий. В первую очередь, создаётся дополнительный excel документ, который в дальнейшем будет именоваться, как «*конфиг*». В данные документ заносится информация, которая необходима для работы робота, но может быть изменена сторонними действиями. Для данной задачи конфиг может содержать следующую информацию (Рисунок 3.1):

- Относительные пути к необходимым папкам (путь к папке с необработанными файлами, путь к паке с результатом обработки и т.д.)
- Регулярные выражения для обработки текста PDF-документов и названия групп в них.



3.1. Пример конфига для робота.

Помимо конфига необходимо убедится в наличие шаблона excel таблицы, куда будут занесены считанные данные. Пример такой таблицы можно увидеть на рисунке 3.2.

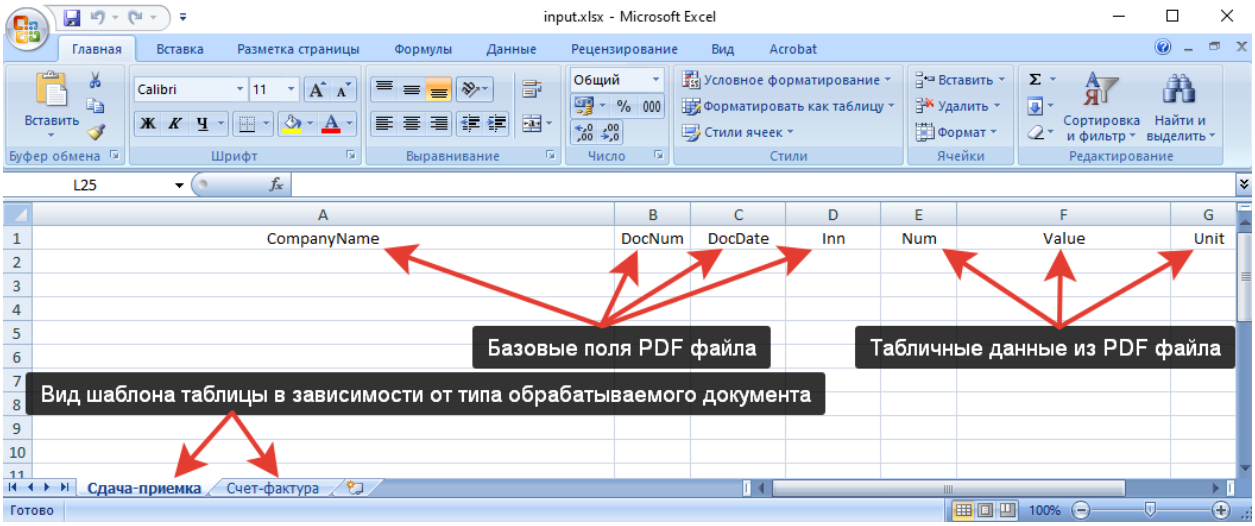


Рисунок 3.2. Пример шаблона excel таблицы.

Так же стоит написать регулярные выражения для обработки считанного из PDF текста. Примеры регулярных выражений для обработки текста и таблицы из PDF документа представлены на рисунке 3.3.

Для написания регулярных выражений можно использовать сайт *«regex101.com»* или какой-нибудь подобный.

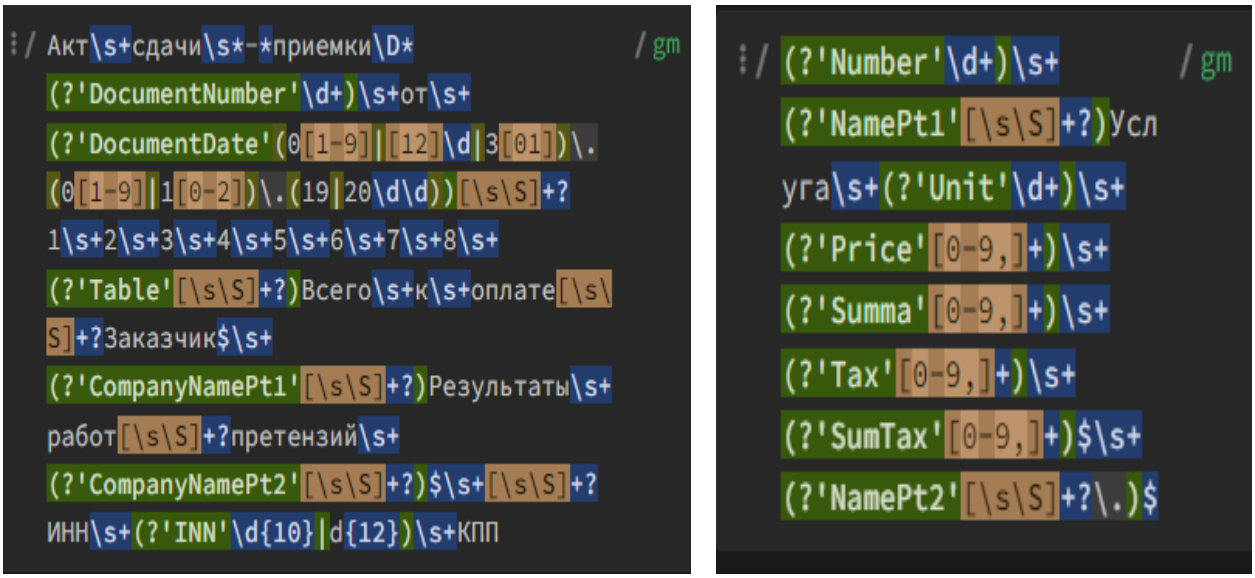


Рисунок 3.3 Пример регулярных выражений для обработки текста (слева) и таблицы (справа)

4. Работа с PIX Studio

Теперь необходимо запустить PIX Studio и создать новый проект. Для этого сначала необходимо нажать на панели слева на иконку блокнота со звёздочкой в левом нижнем углу. В появившемся окне выбирается пункт проект (Рисунок 4.1). PIX Studio попросит ввести название проекта и его расположение.

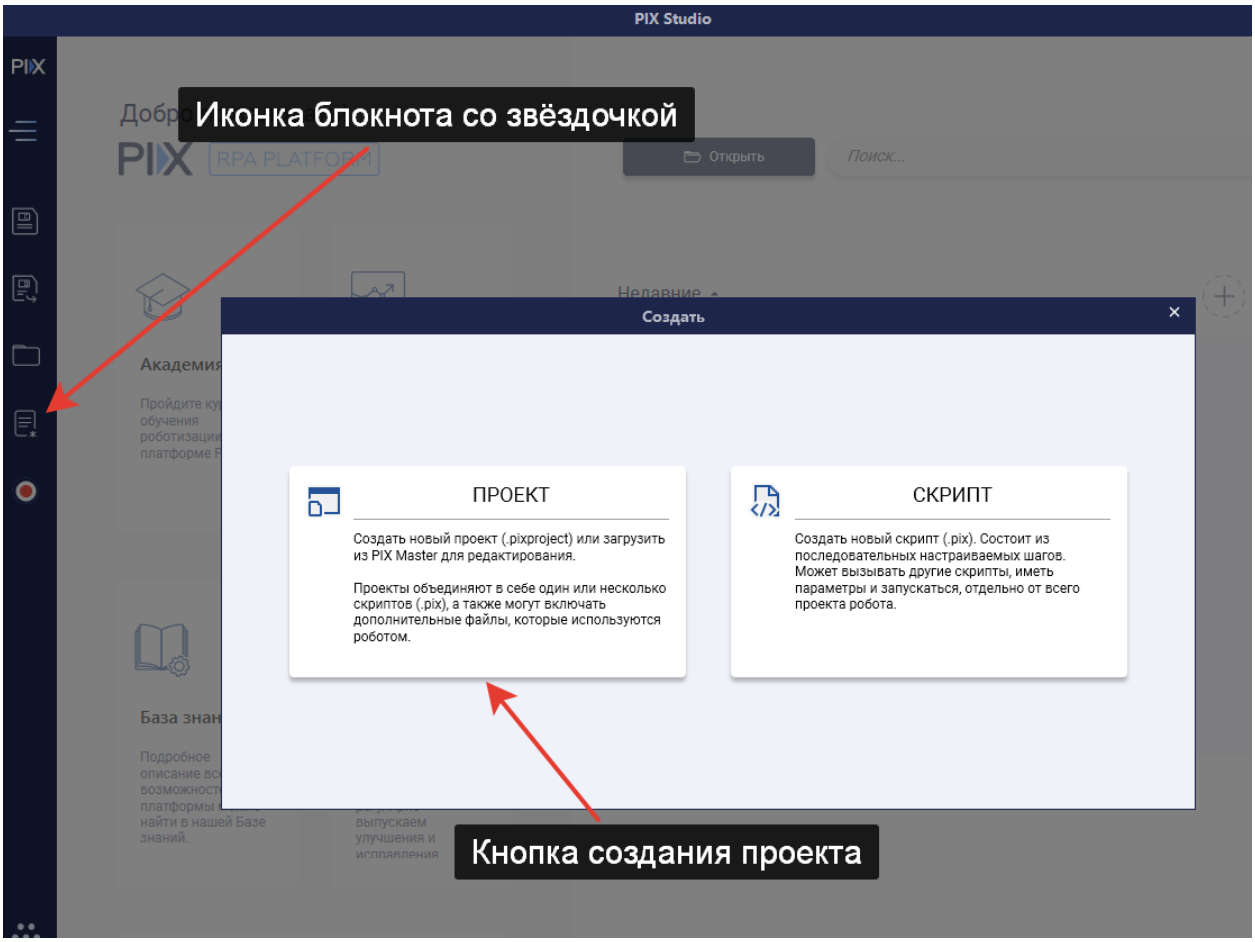


Рисунок 4.1. Создание проекта

4.1. Чтение конфига и создание папок

Затем с помощью активностей *«Базовые/Присвоить значение»* и C# функции new инициализируется словарь для хранения данных из конфига.

Потом с помощью активности *«Office/Excel/Прочитать диапазон ячеек»* считываются данные из конфига и заносятся во временную таблицу. В свойствах активности указывается: *Лист* – название листа в excel документе, с которого нужно прочитать данные, *Диапазон* – диапазон ячеек, который необходимо считать, в случае, если надо считать все непустые ячейки

указывается «””», *С заголовками* – ставится галочка, если в конфиге имеются заголовки, *Путь к файлу* – путь excel файл с конфигом, *Таблица* – переменная, в которую будет сохранена вся информация.

Далее с помощью активности *«Базовые/Присвоить значение»* и C# функции *Таблица.AsEnumerable().ToDictionary<DataRow, Тип_данных_ключа, Тип_данных значения>(Соотношение ключа и значения со столбцами)* таблица конфига преобразуется в словарь. (Рисунок 4.1.1)

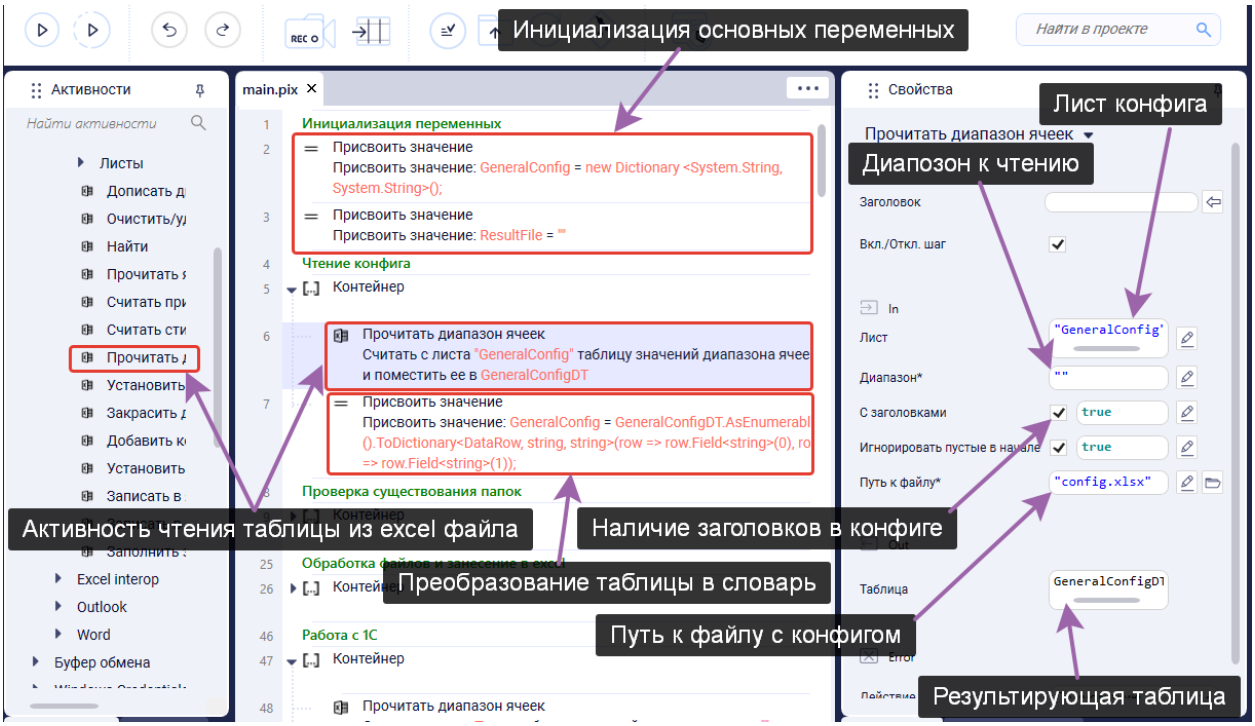


Рисунок 4.1.1. Чтение конфига.

Следующим шагом с помощью активности *«Файлы/Путь существует?»* проверяется наличие необходимой папки. В свойствах указывается: *Путь* – путь к необходимой папке, хранится в словаре конфига, *Результат* – булева переменная, в которую будет записан результат проверки.

Затем с помощью активностей *«Базовые/Условный оператор»* и *«Файлы/Создать папку»* создаётся папка в случае её отсутствия.

Данные действия проводятся для всех необходимых папок. (Рисунок 4.1.2)

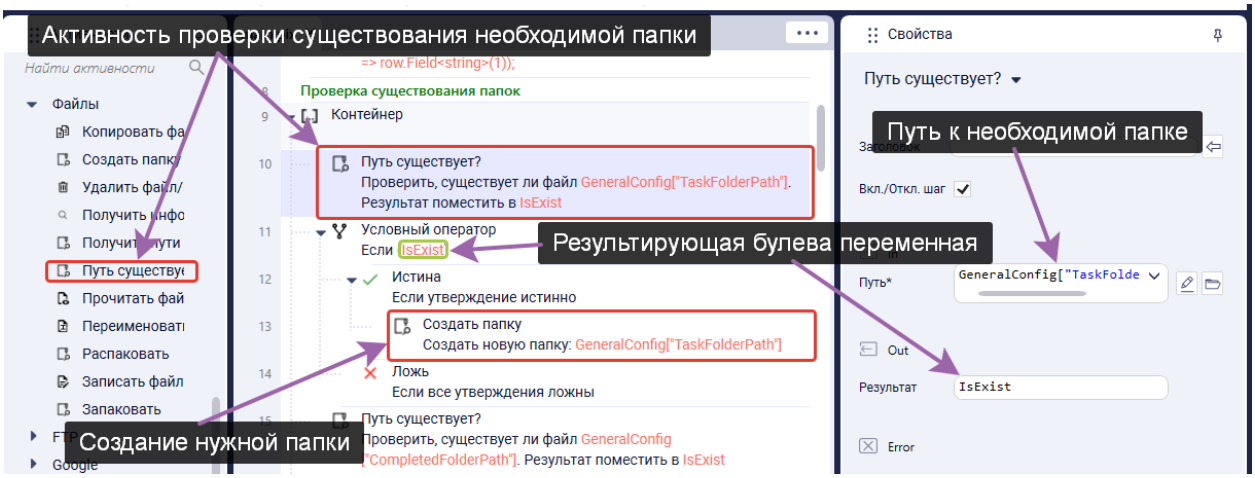


Рисунок 4.1.2. Создание необходимых папок

4.2. Обработка файлов

Сначала с помощью активности *«Коллекции/Таблица/Создать таблицу»* создаётся таблица, в которую будут помещены результаты обработки. В свойствах указываются: *Таблица* – переменная для результирующей таблицы, *Структура таблицы* – создаются необходимые столбцы.

Также с помощью активности *«Коллекции/Словарь/Создать словарь»* создаётся словарь для хранения нужных значений в ходе обработки файла.

Следующим шагом с помощью активности *«Файлы/Получить пути к файлам/каталогам»* получаются пути ко всем ещё необработанным файлам. (Рисунок 4.2.1)

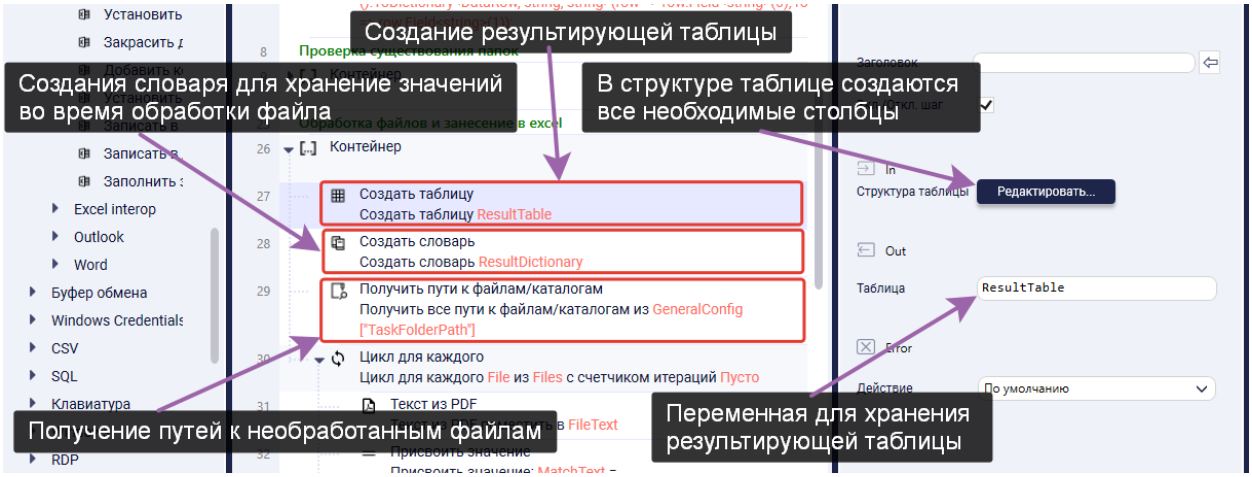


Рисунок 4.2.1. Создание необходимых переменных

Далее с помощью активности *«Базовые/Циклы/Цикл для каждого»* запускается цикл по всем файлам.

В начале цикла с помощью активности *«PDF/Текст из PDF»* достаётся текст из PDF файла. В свойствах указывается: *Файл PDF* – путь до PDF файла, *Результат* – переменная для хранения полученного текста.

Потом с помощью активностей «Базовые/Присвоить значение» и C# функций System.Text.RegularExpressions.Regex.Match(Текст, Регулярное_выражение, Опции_Регулярного_выражения) и System.Text.RegularExpressions.Regex.Matches(Текст_таблицы, Регулярное_выражение, Опции_Регулярного_выражения) к полученному тексту и тексту считанной таблицы применяется регулярное выражение. (Рисунок 4.2.2)

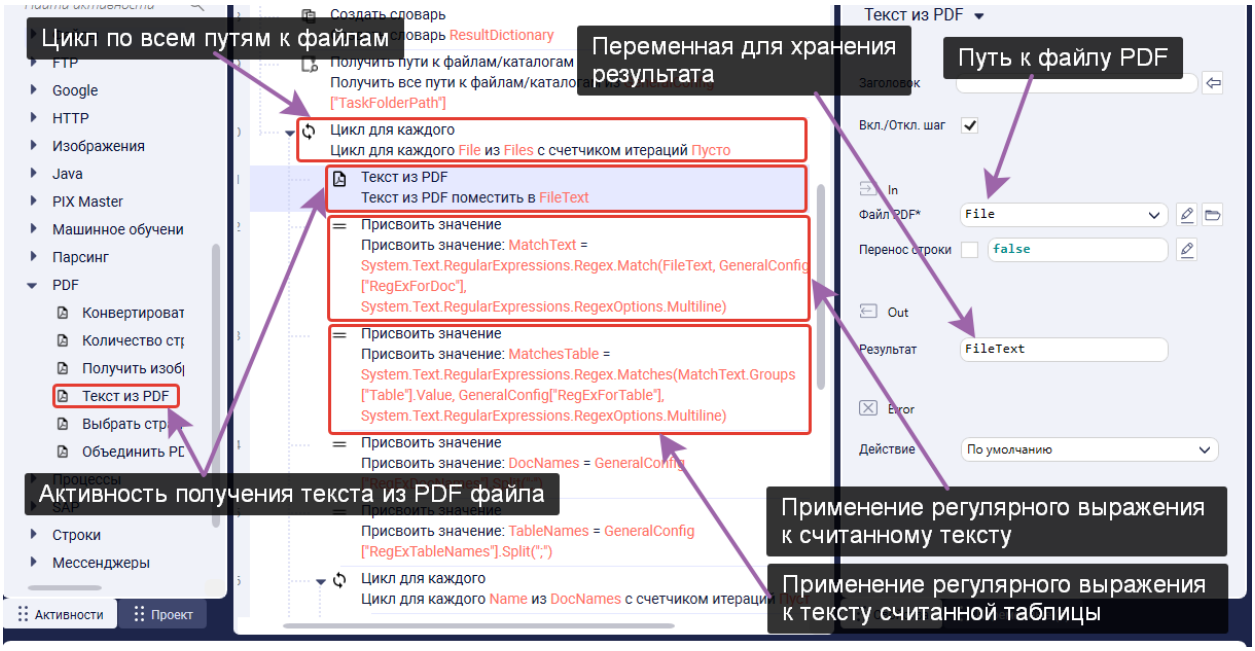


Рисунок 4.2.2. Применение регулярных выражений

Затем с помощью активности *«Базовые/Присвоить значение»* и C# функции *Строка.Split()* в переменные считываются названия групп регулярного выражения.

Затем с помощью активностей *«Базовые/Циклы/Цикл для каждого»* и *«Коллекции/Словарь/Задать значение для ключа»* и C# функции *Результат_регулярного_выражения.Groups[Название_группы]*.Value данные из считанного текста заносятся в словарь с ключом, который равен названию колонки в результирующей таблице. (Рисунок 4.2.3)

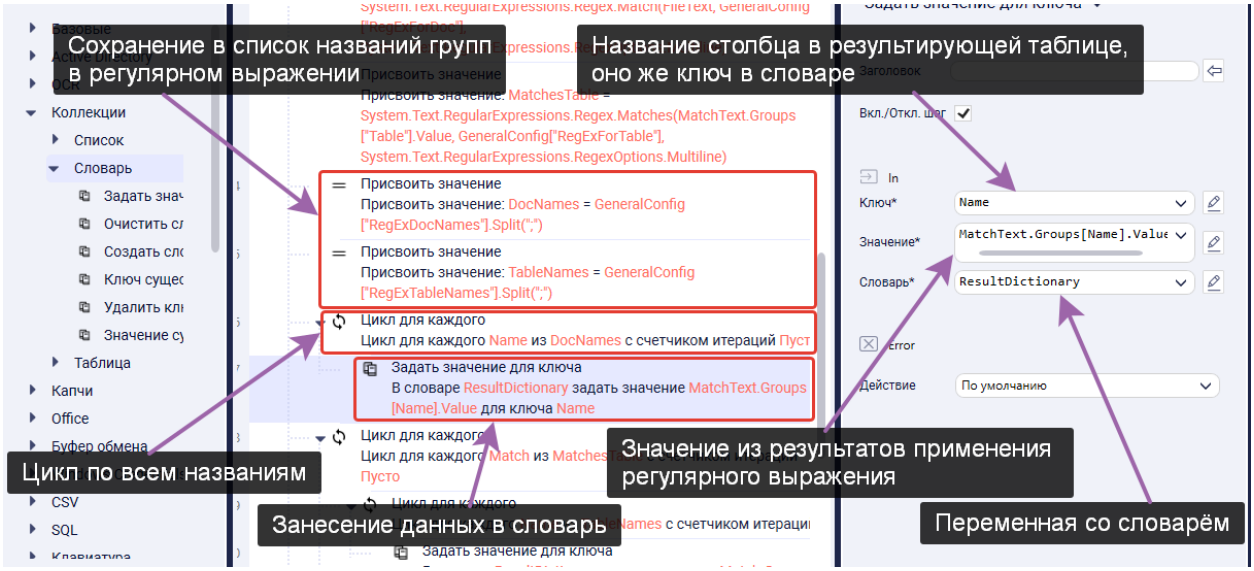


Рисунок 4.2.3. Занесение общих данных из файла в словарь

Далее с помощью активности *«Базовые/Циклы/Цикл для каждого»* запускается цикл по всем найденным строкам таблицы из текста. Для каждой строки аналогичным образом данные заносятся в словарь.

После занесения данных из строки в словарь с помощью активности *«Коллекции/Таблица/Добавить строку»* значения словаря заносятся в результирующую таблицу. В свойствах активности указывается: *Строка* – словарь с данными, где в качестве значений выступают значения ячеек, а в качестве ключей название столбцов, *Позиция* – дописать строку в конец таблицы или в начало, *Таблица* – переменная с результирующая таблицей.

Далее после обработки всех строк таблицы с помощью активности *«Базовые/Присвоить значение»* создаётся полный путь к excel файлу, куда необходимо записать данные.

Следующим шагом с помощью активности *«Файлы/Копировать файл/папку»* файл шаблона копируется в нужное место и переименовывается, а также обрабатываемый файл копируется в папку с обработанными файлами.

В скопированный файл *«Office/Excel/Дописать диапазон»* результирующая таблица записывает в excel файл. В свойствах активности указывается: *Лист* – лист, на который нужно записать данные, *Таблица* – переменная с результирующей таблицей, *Добавить заголовки* – убирается

галочка, так как заголовки уже есть в шаблоне, *Путь к файлу* – путь к конечному excel файлу. (Рисунок 4.2.4)

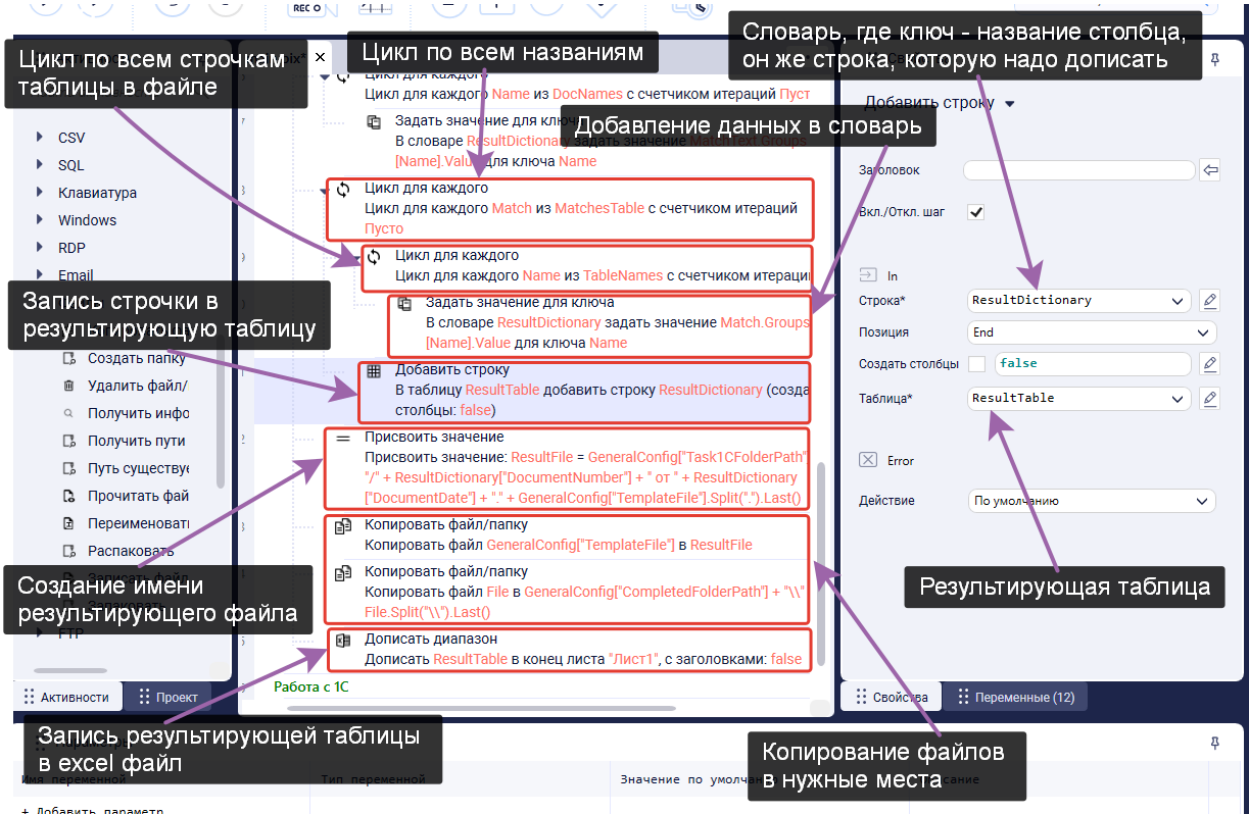


Рисунок 4.2.4. Запись данных в excel файл

4.3. Работа с 1С

Сначала данные из excel документа считываются в таблицу с помощью активности *«Office/Excel/Прочитать диапозон ячеек»*, причём включая заголовки. А также с помощью активности *«Коллекции/Словарь/Создать словарь»* создаётся словарь для хранения значений одной строчки таблицы.

Следующий шагом активностью *«Базовые/Циклы/Цикл для каждого»* запускается цикл по всем строчкам таблицы со счётчиком итераций, чтобы различать заголовок и просто строку. С помощью активности *«Базовые/Условный оператор»* и номера итерации строчка проверяется на заголовок. Если обрабатывается строчка, не являющаяся заголовком, то активностью *«Базовые/Циклы/Цикл для каждого»* запускается цикл по ячейкам, где значение ячейки заносится в словарь с ключом, равным значению соответствующей ячейки заголовка, с помощью активности *«Коллекции/Словарь/Задать значение для ключа»*. (Рисунок 4.3.1)

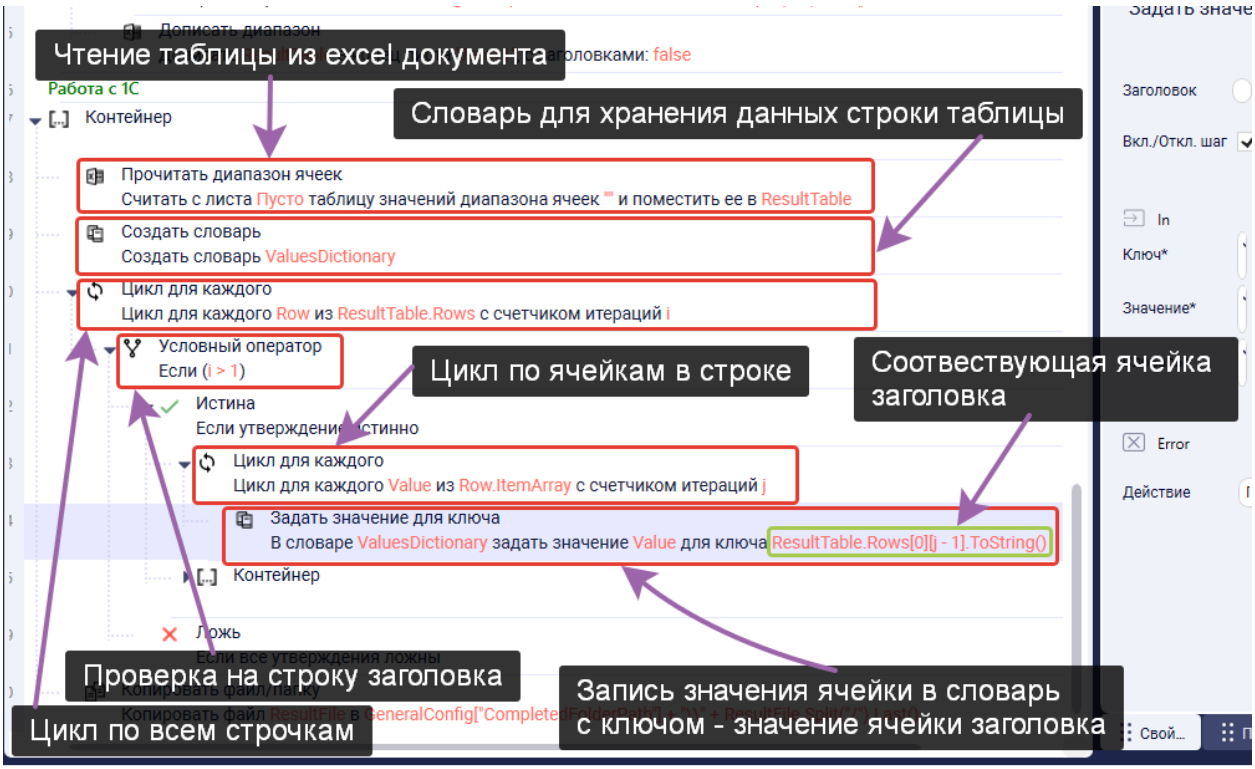


Рисунок 4.3.1. Построчное чтение таблицы из excel файла

Для работы с 1С используется имитация деятельности человека с помощью активностей *«Windows/Клик по UI-элементу»*, *«Window/Впечатать в UI-элемент текст»* и *«Клавиатура/Нажать hotkey»*. В активности нажатия клавиши в свойствах указывается: Клавиша – клавиша, которую необходимо нажать, в данной задаче клавиша *«Enter»*. В активностях с UI-элементом для выбора UI-элемента в свойстве *«XPath»* нажимается значок мышки с салютом на кончике и мышкой выбирается элемент, на который нужно нажать или куда нужно впечатать текст. (Рисунок 4.3.2)

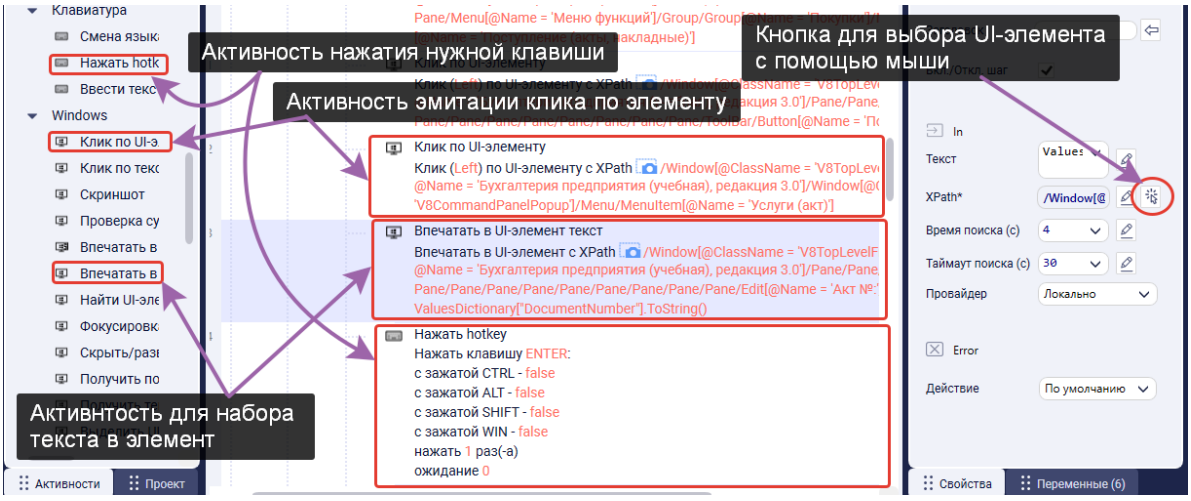


Рисунок 4.3.2. Имитация действий человека.

Литература

1. https://regex101.com/
2. https://metanit.com/sharp/tutorial/16.4.php
3. https://learn.microsoft.com/ru-ru/dotnet/api/system.data.datatable?view=net-5.0
4. https://learn.microsoft.com/ru-ru/dotnet/api/system.collections.generic.dictionary-2?view=net-5.0
5. https://learn.microsoft.com/ru-ru/dotnet/api/system.text.regularexpressions.match?view=net-6.0
6. https://knowledgebase.pixrpa.ru/actions/Desktop/EnterTextElement
7. https://knowledgebase.pixrpa.ru/actions/Desktop/KeyboardSendHotkey
8. https://knowledgebase.pixrpa.ru/actions/Desktop/ClickElement
9. https://knowledgebase.pixrpa.ru/actions/Base/LoopForEach
10. https://knowledgebase.pixrpa.ru/actions/Base/If
11. https://knowledgebase.pixrpa.ru/actions/Dictionary/CreateDictionary
12. https://knowledgebase.pixrpa.ru/actions/Excel/ReadRange
13. https://knowledgebase.pixrpa.ru/actions/Files/CopyFileCatalog
14. https://knowledgebase.pixrpa.ru/actions/Excel/AppendRange
15. https://knowledgebase.pixrpa.ru/actions/DT/AddRow
16. https://knowledgebase.pixrpa.ru/actions/Base/Assign
17. https://knowledgebase.pixrpa.ru/actions/PDF/ReadPDF
18. https://knowledgebase.pixrpa.ru/actions/Files/GetListFilesOrCatalogs
19. https://knowledgebase.pixrpa.ru/actions/DT/CreateTable
20. https://knowledgebase.pixrpa.ru/actions/Files/PathExist
21. https://knowledgebase.pixrpa.ru/actions/Files/CreateFolder