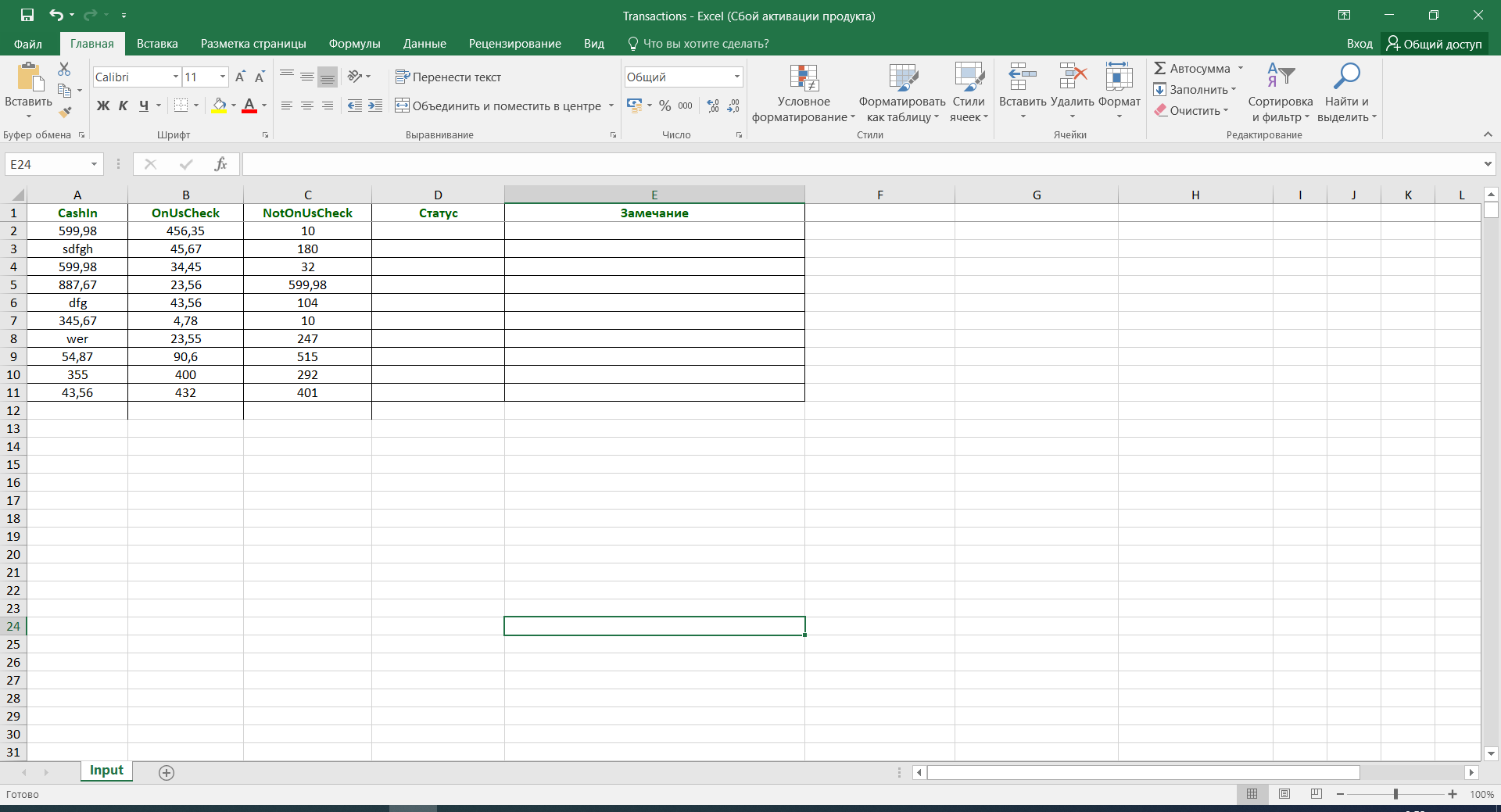
**Задание 6-1**. Создать процесс автоматизации, который откроет приложение и войдет в систему. Затем процесс автоматизации должен обработать транзакции, взятые из определенного Excel-файла, и записать статус и пояснение в тот же Excel-файл рядом с соответствующей транзакцией.

Чтобы продемонстрировать транзакций в данном примере используется приложение **UIDemo.exe**. При его запуске появляется окно, в котором надо ввести имя **admin** и пароль **password**, нажать кнопку **Log In** и появится окно **UiDemo**. В этом окне будет демонстрироваться выполнение транзакций.

|  |  |
| --- | --- |
|  |  |

Исходные данные для транзакций находятся в файле **Transactions.xlsx**:



В UiPath Studio создать новый процесс, используя **шаблон** Robotic Enterprise Framework (Старт/Роботизированная инфраструктура предприятия).

В папку **Data** поместить Excel-файл **Transactions.xlsx** с описанием транзакций и файл **UIDemo.exe**, который при запуске проверяет имя и пароль пользователя и открывает окно приложения, демонстрирующего выполнение транзакций.

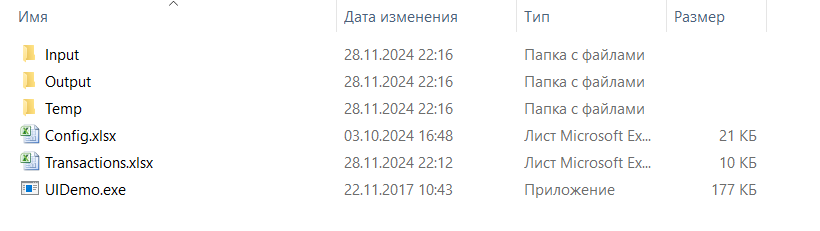


Рисунок 6.1 – Папка Data

Открыть конфигурационный файл **Data/Config.xlsx**.

На листе **Settings** изменить значение параметра **logF\_BusinessProcessName** на **RePrimer** и добавить следующие настройки:

|  |  |  |
| --- | --- | --- |
| **Name** | **Value** | **Description** |
| UIDemo\_ApplicationPath | Data\UIDemo.exe | UIDemo app path |
| UIDemo\_InputFile | Data\Transactions.xlsx | Path to excel file with transactions |
| Input\_SheetName | Input | Name of sheet with transaction data |

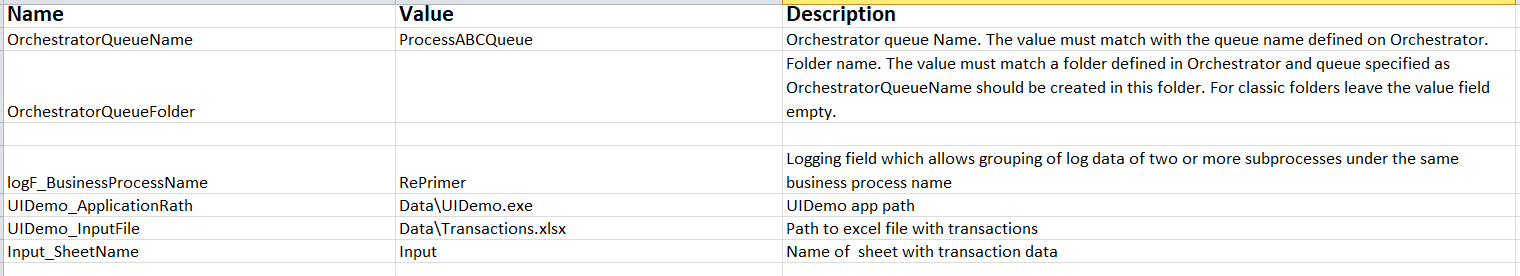


Рисунок 6.2 – Лист Settings

На листе **Constants** изменить значение параметра **MaxRetryNumber** на **1**. Сохранить файл.

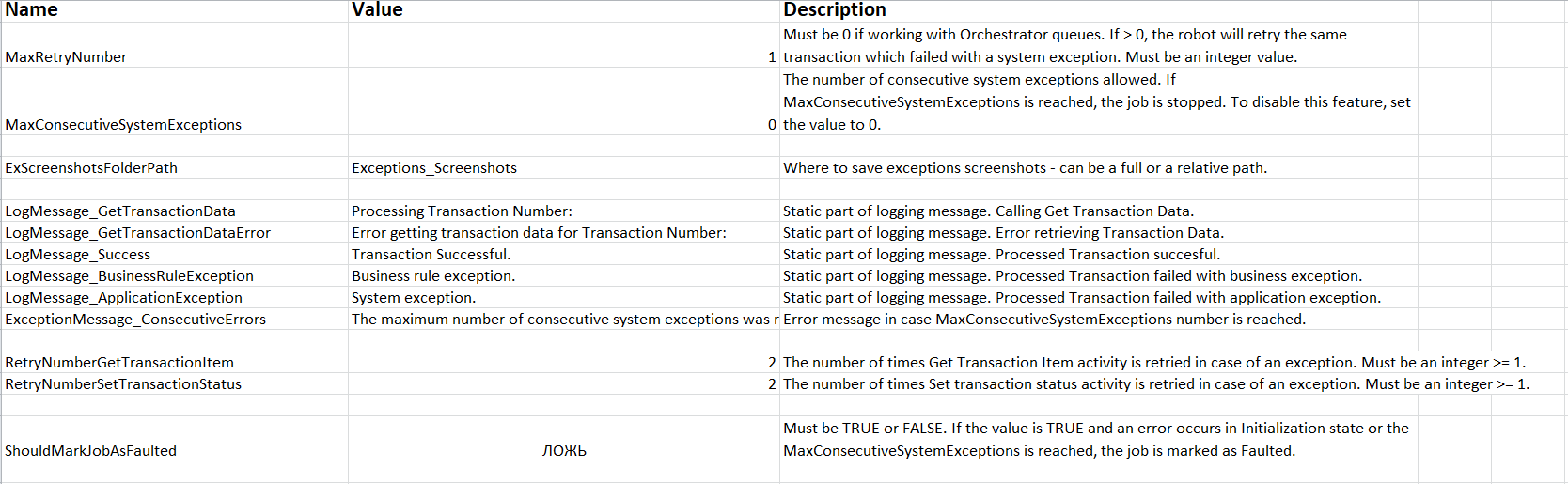


Рисунок 6.3 – Лист Constants

Открыть рабочий процесс **Main.xaml** и раскрыть состояние **Init** (Инициализация).

Перенести действие **Sequence** после **Invoke KillAllProcesses workflow** и назвать его **Добавление элементов в таблицу данных** (**Add Items to DataTable)**.

Добавить в последовательность действие **Read Range** из раздела **Рабочий журнал** (Workbook**)** и назвать **Считывание входных данных** (**Read Input Data Range)**.

Установить следующие свойства:

|  |  |
| --- | --- |
| **Свойство** | **Значение** |
| WorkbookPath | Config("UIDemo\_InputFile").ToString |
| SheetName | Config("Input\_SheetName").ToString |
| Range |  |
| DataTable | TransactionData |

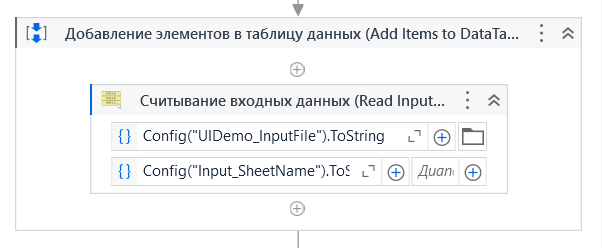


Рисунок 6.4 – Действие Read Range

Добавить действия **Log Message** до и после действия **Read Range.**

В первом указать в поле **Message** значение **"Чтение входных данных из Excel"**, во втором – **"Записи добавлены в DataTable"**.

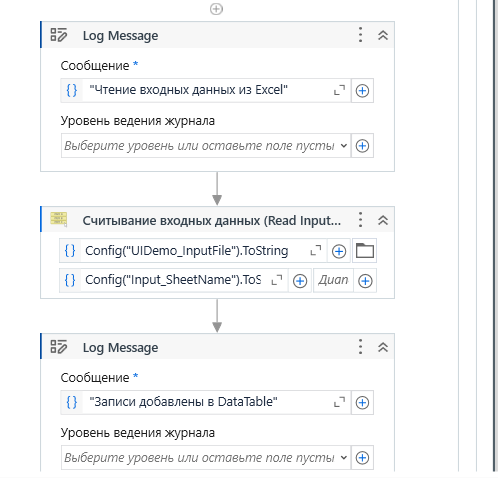


Рисунок 6.5 – Действие Log Message

На панели переменных поменять значение переменной **TransaсtionNumber** на **0**, а также изменить тип переменной **TransactionItem** на **DataRow**.

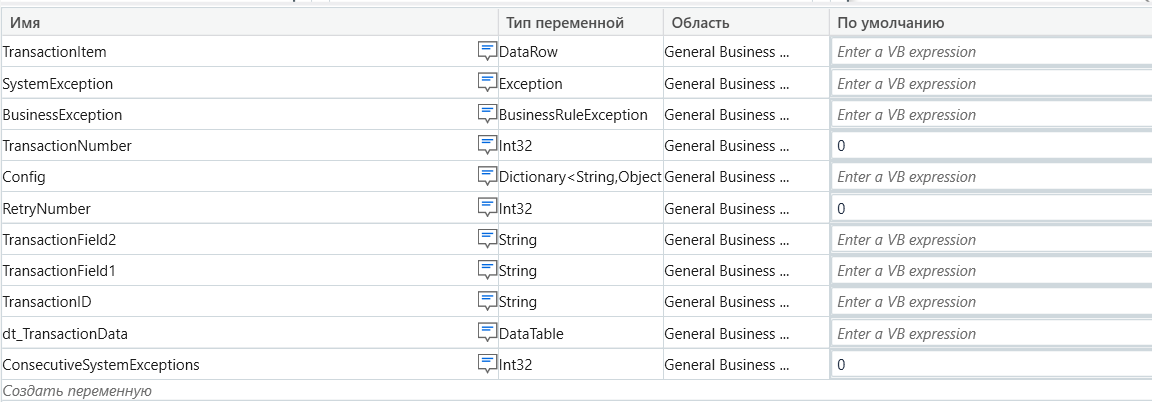


Рисунок 6.6 – Окно переменных

Вернуться в рабочий процесс **InitAllApplications.xaml**.

Добавить действие **Open Application**.

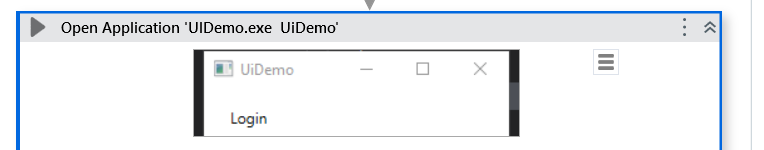


Рисунок 6.7 – Действие Open Application

Открыть рабочий процесс **Framework/KillAllProcesses.xaml**.

Перенести два действия **Kill Process**.

В первом действии в свойстве **ProcessName** указать **"excel"**, во втором – **"UIDemo"**.

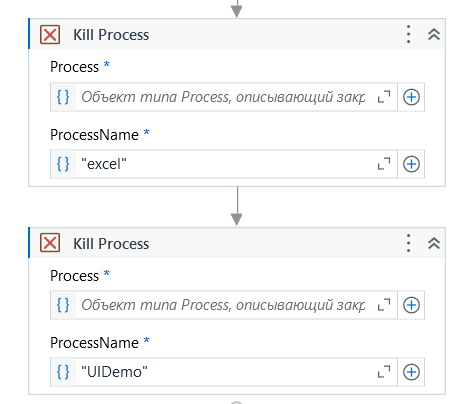


Рисунок 6.8 – Действия Kill Process

Открыть рабочий процесс **Framework/GetTransactionData.xaml.**

На панели **Аргументы** изменить тип аргумента **out\_TransactionItem** на **DataRow**.

Удалить из рабочего процесса действие **Get Transaction Item**.

В действии **If a new Item is fetched Grab it's significant data for logging** изменить условие **Condition** на **in\_TransactionNumber < io\_TransactionData.Rows.Count**.

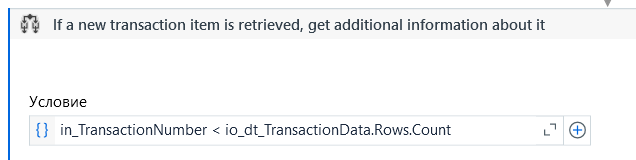


Рисунок 6.9 – Действия If

В полях указываем необходимые переменный. В поле слева указать переменную **out\_TransactionItem**, а в поле справа ввести **io\_TransactionData(in\_TransactionNumber).**

В блоке **Then** в правом поле действия **Assign TransactionField1** изменить значение на **out\_TransactionItem.Item("CashIn").ToString**.

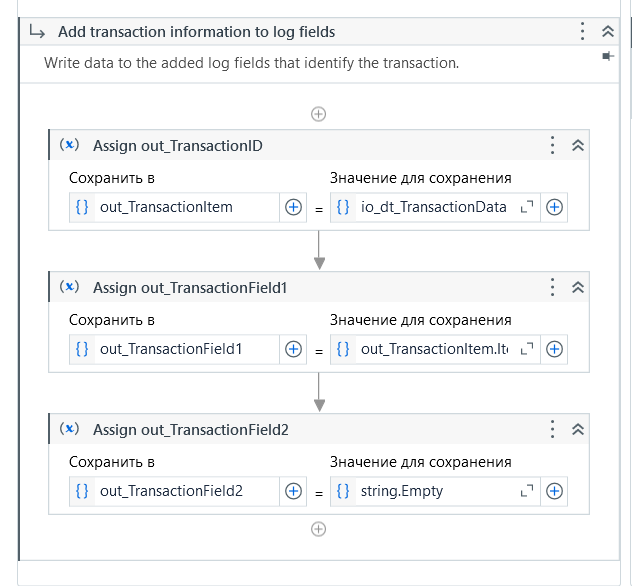


Рисунок 6.10 – Действия Assign

В блок **Else** перенести действие **Assign** и установить значение аргумента **out\_TransactionItem** в **Nothing**.

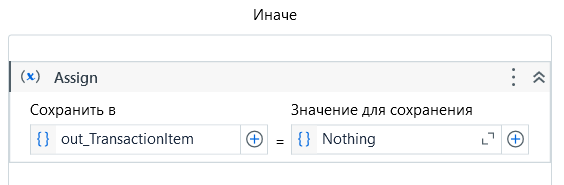


Рисунок 6.11 – Блок Else

Вернуться в рабочий процесс **Main.xaml**, раскрытьсостояние **Get Transaction Data** (Получение данных), перейти к моменту вызова рабочего процесса **Framework/GetTransactionData.xaml**

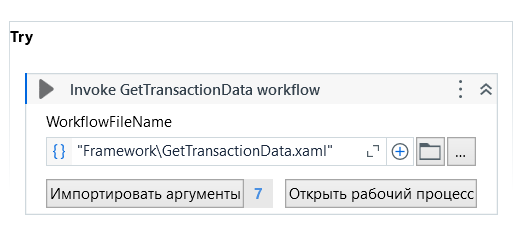


Рисунок 6.12 – Действие Invoke Workflow

В действии **Invoke** **GetTransactionData** **workflow** нажать кнопку **Импортировать аргументы**, добавить в значение аргумента **out\_TransactionItem** переменную **TransactionItem** и изменить тип аргумента на **DataRow**.

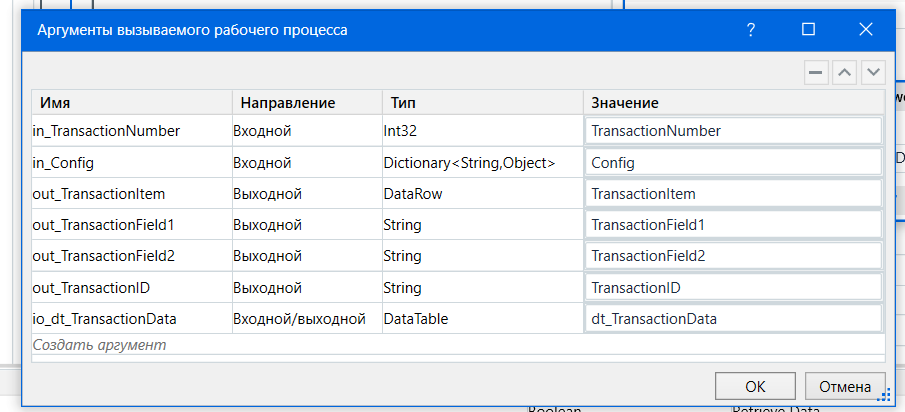


Рисунок 6.13 – Импорт аргументов

Добавить действие **If** в последовательность.

Ввести в свойство **Condition** условие: **IsNumeric(in\_TransactionItem("Cash-In")) AndAlso IsNumeric(in\_TransactionItem("OnUsCheck")) AndAlso IsNumeric(in\_TransactionItem("NotOnUsCheck"))**.

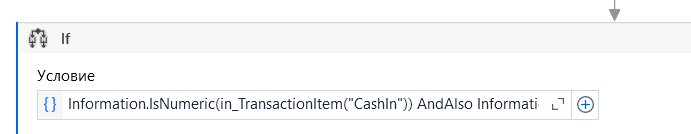


Рисунок 6.14 – Действия If

Добавление нескольких действий Type Info со своими настройками

Перенести действие **Type Into** в последовательность **Do**.

Нажать в действии на ссылку **Указать на экране dow**, щелкнуть в приложении по полю **Сash In**, в свойство **Text** добавить **in\_TransactionItem("CashIn").ToString**, поставить галочку в свойстве **Simulate Type**.

Перенести действие **Type Into**.

Нажать в действии на ссылку **Указать на экране**, щелкнуть в приложении по полю **On Us Check**, в свойство **Text** добавить **in\_TransactionItem("OnUsCheck").ToString**, поставить галочку в свойстве **Simulate Type**.

Добавить действие **Type Into**.

Нажать на ссылку **Указать на экране**, щелкнуть в приложении по полю **Not** **On Us Check.**

В свойство **Text** добавить **in\_TransactionItem("NotOnUsCheck").ToString**, поставить галочку в свойстве **Simulate Type**.

Добавить действие **Click**,

Нажать в действии на ссылку **Указать на экране**, щелкнуть в приложении по кнопке **Accept**, поставить галочку в свойстве **Simulate Click**.

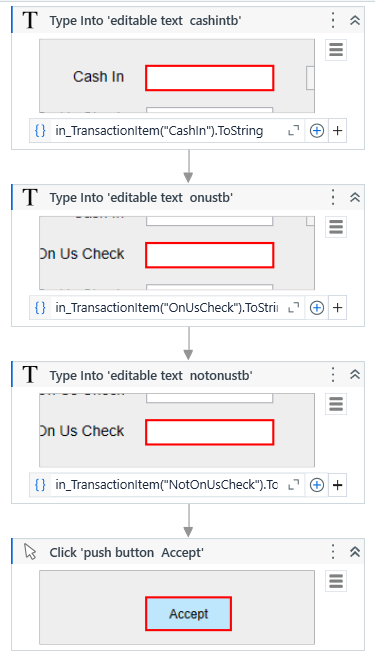


Рисунок 6.15 – Содержимое блока Do

В блок **Else** перенести действие **Throw**,

В свойстве **Exception** указать **new BusinessRuleException("**Записи не числовые - обработка невозможна**")**.

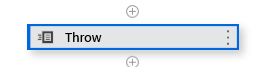


Рисунок 6.16 – Действия Throw

Открыть процесс **Framework/SetTransactionStatus.xaml**.

На панели аргументов изменить тип аргумента **in\_TransactionItem** на **DataRow**.

Перейти в последовательность **Success**.

В действии **If** в поле **Condition** поменять **Uipath.Core.QueueItem** на **System.Data.DataRow**.

В блоке **Then** удалить действие **Set Transaction Status.**

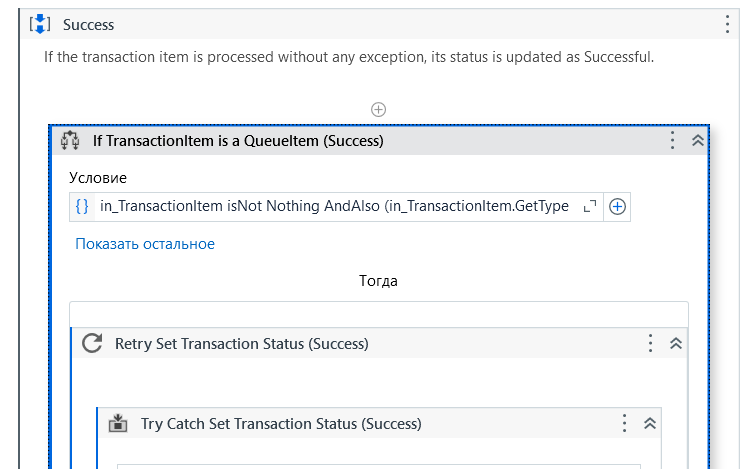


Рисунок 6.17 – Success sequence

При запуске процесса он самостоятельно запустит приложение и войдёт в его систему. После чего начнёт самостоятельно заполнять форму данными из файла. Все этапы работы приложения можно просмотреть в окне вывода.

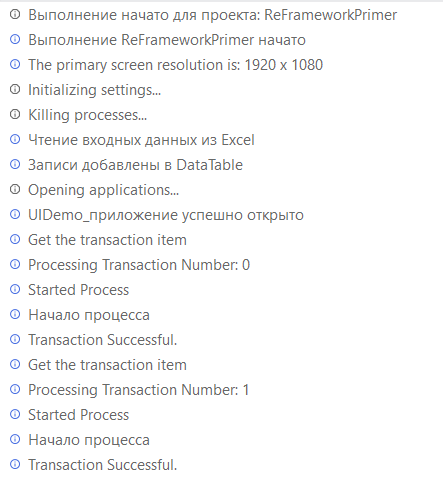


Рисунок 6.18 – Окно вывода