

Лабораторная работа № 1. Ознакомление с UiPath Studio

Описание интерфейса UiPath Studio приведено в [приложении](#) в конце данного практикума.

Разработка проекта

При разработке проекта используются понятия:

1. **Activity (Действие)** – любая небольшая задача, которая выполняется в UiPath Studio, например, щелчок мыши, нажатие клавиши на клавиатуре, копирование текста откуда-то и т. д. Если нужное действие не находится на экране, то можно его найти с помощью **поиска** на панели **Действия** в левой части экрана.
2. **Sequence (Последовательность)** – последовательность действий, которые ведут к решению задачи.

Задание 1-1. Создать робот в UiPath Studio, который выведет слово **Hello ...** в окне сообщения **Message Box**, используя действие **Sequence**.

1. Открыть UiPath Studio, выбрать пункт **Процесс**, в появившемся окне выбрать место расположения робота, назвать его, например, **Hello-1**. Можно ввести краткое описание проекта в поле **Description (Описание)** и нажать кнопку **Create (Создать)**. Откроется главное окно проекта.
2. В нижней части экрана слева открыть панель **Activities (Действия)**. Перетащить действие **Sequence (Последовательность)** из этой панели в окно рабочего процесса в центре экрана. Щелкнуть левой клавишей мыши по заголовку этого действия и изменить имя с **Sequence** на **Sequence - 'Вывод текста в окно сообщений'**.
3. Создать две переменные на панели **Variables (Переменные)**: переменную **Result** (значение по умолчанию – **"Hello World"**) и **Name**, обе типа **String**.
4. Перетащить действие **Input Dialog** в окно **Sequence**. В правой части экрана на панели **Properties (Свойства)** для этого действия ввести в поля следующие значения:

Label	Title	Result
"Enter Details"	"What is your name?"	Name

5. Перетащить действие **Assign**. Указать для этого действия переменную **Result** в поле **To** и **"Hello, "+ Name** в поле справа от знака равенства.
6. Перетащить действие **Message Box** в **Sequence**. В свойстве **Text** этого действия указать переменную **Result**.
7. Просмотреть всю последовательность действий и свойства. Если в правом верхнем углу элемента имеется синий кружок с восклицательным знаком внутри, значит этот элемент или свойство содержит *ошибку*. Надо поместить курсор на кружок, прочитать комментарий к ошибке и исправить ее.
8. Запустить робот на выполнение, нажав кнопку **Запустить** в верхней части экрана.

Задание 1-2. Создать робот в UiPath Studio, который выведет слово **Привет ...** в окне сообщения **Message Box** и в информационном окне **Output**, используя действие **Flowchart (Блок-схема)**.

1. Открыть **UiPath Studio**, создать процесс и назвать его **Hello-2**.
2. Перетащить действие **Flowchart** из панели **Activities (Действия)** и поместить его в рабочий процесс. Появится элемент **Start**.

3. Создать две переменные на панели **Variables (Переменные)**: переменную **Result** (значение по умолчанию – "**Привет мир!**") и **Name**, обе типа **String**.

4. Добавить действие **Input Dialog** в **Flowchart**, поместив его на нижний треугольник (действие соединится с элементом **Start**) или соединить центр нижней границы этого действия с центром верхней границы элемента **Start** протяжкой мыши. На панели **Properties** этого действия ввести следующие значения:

Title	Result
"Ваше имя?"	Name

5. Добавить действие **If** в **Flowchart**. Соединить его с действием **Input Dialog** протяжкой мыши. Внутри операции **If** написать условие (condition) **Name <> ""**.

В разделе **Then** добавить действие **Assign**. Указать для этого действия переменную **Result** в поле **To** и "**Привет,** " + **Name** в поле справа от знака равенства.

В контекстном меню заголовка действия **If** выполнить команду **Показать родительский элемент** или щелкнуть мышью по слову **Main** сверху.

6. Добавить действие **Message Box** в **Flowchart**, соединить его с действием **If**. В свойстве **Text** этого действия указать переменную **Result**.

7. Добавить действие **Write Line**, соединить его с **Message Box**. В свойстве **Text** этого действия указать переменную **Result**.

8. Запустить робот на выполнение (окно **Выход** находится в нижнем левом углу экрана).

9. Добавить ввод фамилии и организовать приветствие по имени и фамилии.

Задание 1-3. Создать робот в UiPath Studio, который вводит текст в приложении Блокнот и меняет его шрифт. Использовать режим **базовой** записи (**Basic**).

1. Открыть приложение **Блокнот**.

2. В UiPath Studio создать проект и назвать его **Recorder**.

3. Создать последовательность **Sequence**.

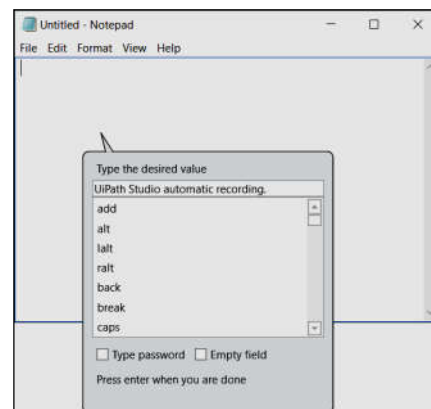
4. На ленте **Проект** выбрать **Запись**, а затем в раскрывающемся списке тип записи: **Базовая (Basic)**. Нажать в появившемся окне **Запись (Record)**. Начнется процесс автоматической записи.

5. В Блокноте нажать на главную панель. Отобразится всплывающее окно. Выбрать нужный пункт, например, **add**, ввести в верхней строчке любой текст и нажать клавишу **Enter**. Строка отобразится в Блокноте. (Если выбран флажок **Пустое поле (Empty field)**, то удаляется ранее существующий текст).

5. Выбрать меню **Формат (Format)**, на вопрос о привязке ответить **Нет**, выбрать **Шрифт (Font)** и изменить его, например, на полужирный курсив.

6. Нажать клавишу **Esc** два раза. Прекращается запись. Проект надо сохранить и закрыть приложение Блокнот без сохранения содержимого.

7. Вновь открыть Блокнот. Запустить робот на выполнение.



Лабораторная работа № 2. Выбор направлений выполнения проекта

Для выбора направлений выполнения проекта используется действие **If**, действие **Flow Decision**, действие **Switch**.

Задание 2-1. Создать робот, который запрашивает у пользователя два числа, проверяет, делится ли одно из них на другое и, в зависимости от результата, отображает разные сообщения на панели **Выход (Output)**.

1. Создать проект с именем **IfActivity**.

2. Перетащить последовательность **Sequence** на рабочую панель.

Создать две целочисленные переменные **FirstNumber** и **SecondNumber**.

3. Добавить два действия **Input Dialog** на рабочую панель друг под другом.

На панели свойств ввести подписи и заголовки для обоих действий, а в полях **Результат** ввести переменные **FirstNumber** для первого действия и **SecondNumber** для второго.

4. Добавить действие **If**.

В разделе **Condition** ввести **FirstNumber mod SecondNumber = 0**. Это выражение проверяет, делится ли первое число на второе, используя оператор **mod**.

В разделе **Then** добавить действие **Write Line**.

В свойствах в поле **Text** ввести **FirstNumber.ToString + " делится на " + SecondNumber.ToString + "."**. Это сообщение, которое отображается, если первое число делится на второе.

В разделе **Else** добавить еще одно действие **Write Line**.

В поле **Text** ввести **FirstNumber.ToString + " не делится на " + SecondNumber.ToString + "."**. Это сообщение, которое отображается, если первое число не делится на второе.

5. Запустить робот на выполнение, нажав клавишу **F5**.

Задание 2-2. Создать робот, который играет с пользователем на угадывание числа. Число генерируется случайным образом в пределах от 1 до 9.

1. Создать новый процесс и назвать его **IgraFlowDecision**.

2. Перетащить действие **Flowchart** на рабочую панель.

Создать две переменные типа **Int32** с именами **RandomNumber** (для случайного числа) и **GuessNumber** (для ответа пользователя) и одну строковую переменную типа **String** с именем **Message** (для сообщения пользователю) и значением по умолчанию "Введите число от 1 до 9".

3. Добавить действие **Assign** на рабочую панель и соединить его с узлом **Start**.

В поле **To** ввести переменную **RandomNumber**, а в поле **Value** ввести текст **new Random().Next(1,9)**.

4. Добавить действие **Input Dialog** и соединить его с действием **Assign**.

На панели свойств в поле **Label** ввести переменную **Message**, а в поле **Результат** ввести переменную **GuessNumber**.

5. Добавить действие **Flow Decision** и соединить его с действием **Input Dialog**.

На панели свойств в поле **Condition** ввести **GuessNumber = RandomNumber** (для того, чтобы проверить, совпадает ли число, указанное пользователем, со случайно сгенерированным числом).

6. Добавить действие **Message Box** и соединить его с ветвью **True** действия **Flow Decision**.

На панели свойств в поле **Text** ввести текст "Вы угадали! Число = " + **RandomNumber.ToString** + ".". Это сообщение будет отображаться, если пользователь правильно угадал число.

7. Добавить новое действие **Flow Decision** и соединить его с ветвью **False** действия **Flow Decision**.

На панели свойств в поле **Condition** ввести текст **GuessNumber > RandomNumber** (для того, чтобы проверить, является ли число, указанное пользователем, больше, чем случайным образом сгенерированное).

В поле **DisplayName** ввести текст **Comparison**, что позволит различить два использованных действия **Flow Decision**.

8. Добавить действие **Assign** и соединить его с ветвью **True** действия **Flow Decision - Comparison**.

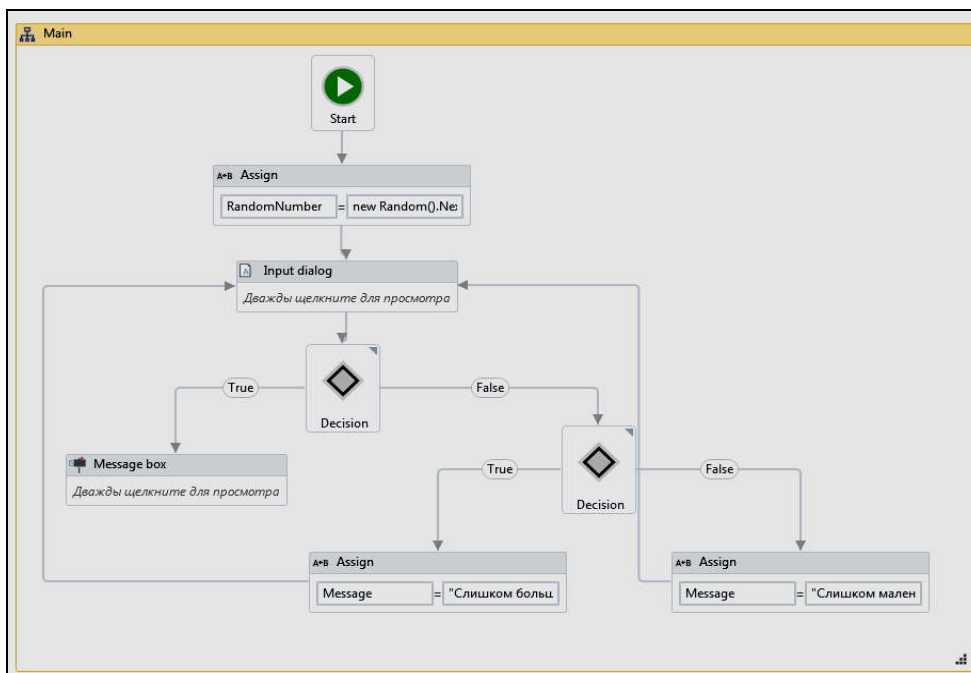
В поле **To** ввести переменную **Message**, а в поле **Value** ввести сообщение, указывающее, что предполагаемое число оказалось больше загаданного, например, "Слишком много. Попробуйте еще".

9. Скопировать действие **Assign** нажатием Ctrl+C, затем CTRL+V. Отобразится дубликат предыдущего действия **Assign**.

Соединить его с ветвью **False** действия **Flow Decision - Comparison** и на панели свойств в поле **Value** ввести текст "Слишком мало. Попробуйте еще".

10. Соединить оба действия **Assign** с действием **Input Dialog** протяжкой мыши. Создается цикл, в котором пользователю предлагается ввести меньшее или большее число, пока он не угадает.

11. Запустить робот на выполнение.



Задание 2-3. Создать робот, который запрашивает у пользователя число, проверяет, четное оно или нечетное и выводит сообщение на панель **Выход (Output)**.

1. Создать проект и назвать его **FirstSwitch**.

2. Создать последовательность **Sequence**.

3. Добавить действие **Input Dialog** на панель конструктора. Создать целочисленную переменную **Number**.

В свойствах ввести в поле **Label** текст “Введите число”. В поле **Результат** ввести переменную **Number**.

4. Добавить действие **Switch**.

В поле **Expression** ввести выражение **Number mod 2**.

В разделе **Default** добавить действие **Write Line**.

В свойствах этого действия в поле **Text** ввести текст **Number.ToString + " является четным числом."**.

Щелкнуть строку **Добавьте новый вариант** и в поле **Case** ввести число 1.

Добавить действие **Write Line**.

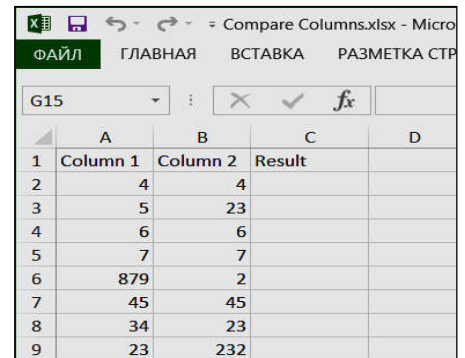
В свойствах этого действия в поле **Text** ввести текст **Number.ToString + " является нечетным числом."**.

5. Запустить робот на выполнение. Опробовать различные варианты ввода чисел.

Лабораторная работа № 3. Обработка информации в приложении Excel и Word

Задание 3-1. Создать робот в UiPath Studio для сравнения двух столбцов в таблице Excel и отображения результата сравнения в третьем столбце.

Создать таблицу Excel с именем **Compare_Columns.xlsx**:



	A	B	C	D
1	Column 1	Column 2	Result	
2	4	4		
3	5	23		
4	6	6		
5	7	7		
6	879	2		
7	45	45		
8	34	23		
9	23	232		

1. Создать процесс и назвать его **CompareColumnsExcel**.

2. Перетащить контейнер **Sequence** из панели **Действия** и поместить его на рабочую панель. Назвать действие **Sequence** как **Sequence - 'Сравнение столбцов'**.

3. Добавить действие **Excel Application Scope**.

В текстовом поле действия в поле **Путь к рабочему...** указать в кавычках имя файла Excel, который необходимо использовать: **"Compare_Columns.xlsx"** (если файл Excel находится не в той же папке, что и проект, то надо указать полный путь к нему).

4. Щелкнуть правой кнопкой мыши по последовательности **Сделать (Do)** и в контекстном меню выбрать **Заметки => Добавить заметку**, ввести текст **Чтение каждой строки Excel-файла и сравнение содержимого столбцов**.

5. Добавить действие **Read Range** в последовательность **Сделать**. Написать название листа Excel, например, **Лист1**.

Создать переменную **DtTable** типа **DataTable** (для выбора типа в поле **Тип переменной => Выбор типов => DataTable**) и указать ее в свойствах в поле **DataTable**.

Создать переменную **RowNumber** типа **Int32** (значение по умолчанию **2**)

6. Добавить действие **For Each Row**.

Указать **row** в секции **For Each** и **DtTable** в свойстве **DataTable**.

7. Добавить действие **If** в последовательность **Body**.

Внутри действия **If** написать условие (Condition) **cint (row (0)) = cint (row (1))**.

В разделе **Then** операции **If** добавить действие **Write Cell**. На панели свойств этого действия ввести следующие значения:

Sheet Name	Range	Value
"Лист1"	"C" + RowNumber.ToString	"Равны"

В раздел **Else** скопировать действие **Write Cell**. На панели свойств в поле **Value** ввести текст **"Не равны"**.

8. Добавить действие **Assign**.

Указать переменную **RowNumber** в поле **To** и **RowNumber + 1** в поле **Value**.

9. Запустить робот на выполнение. Проверить результат в файле Excel.

Задание 3-2. Создать робот в UiPath Studio, который читает данные в таблице Excel с результатами сдачи экзамена на первом листе, определяет и записывает на втором листе, сколько человек получили оценки, больше или равные пяти, и сколько меньше.

Создать файл Excel с именем **Examen.xlsx** примерно такого вида, как на рисунке. Информация (на русском или английском языке) находится на листе с именем **Лист1**.

	A	B	C
1	№	Фино	Оценки
2	1	Иванов	9
3	2	Петров	6
4	3	Сидоров	3
5	4	Кузнецов	8
6	5	Козлов	4
7	6	Смит	6
8			

1. Открыть Studio и создать новый процесс с именем **ResultsExamExcel**. Поместить файл **Examen.xlsx** в папку **ResultsExamExcel**.
2. Перетащить контейнер **Sequence** на рабочую панель.
Создать переменную **excelFile** типа **String**, значение по умолчанию: **"Exam results.xlsx"**, переменную **studentGrades** типа **DataTable** и две переменные типа **Int32**: **failedStud** и **passedStud**.
3. Добавить действие **Excel Application Scope** внутрь последовательности.
В свойствах в поле **WorkbookPath** ввести переменную **excelFile**.
4. Поместить действие **Excel Read Range** внутрь блока **Do**.
В свойствах действия в поле **SheetName** ввести текст **"Лист1"**, в поле **Range** ввести текст **"A1:C7"** и в поле **DataTable** ввести переменную **studentGradesDt** (таблица оценок студентов).
5. Добавить действие **For Each Row** под действием **Build Data Table**.
На панели свойств ввести переменную **studentGrades** в поле **DataTable**.
6. Добавить условие **If** в тело действия **For Each Row**.
Вставить значение **cint(row("Оценки")) < 5** в поле **Condition**. Эта формула анализирует данные, доступные в строке **Grade** из файла **.xlsx**, и решает, значение меньше или больше числа 5.
7. Поместить действие **Assign** в поле **Then**. Добавить переменную **failedStud** в поле **To** и условие **failedStud+1** в поле **Value**.
8. Поместить действие **Assign** в поле **Else**. Добавить переменную **passedStud** в области **To** и условие **passedStud+1** в области **Value**.
9. Поместить действие **Write Cell** под действием **For Each Row**. На панели свойств добавить имя **"Лист2"** в поле **SheetName**, значение **"A1"** в поле **Range** и текст **"Оценки<5"** в поле **Value**.
10. Добавить действие **Write Cell**. На панели свойств ввести имя **"Лист2"** в поле **SheetName**, значение **"A2"** в поле **Range** и **failedStud.ToString** в поле **Value**.
11. Добавить действие **Write Cell**. На панели свойств добавить имя **"Лист2"** в поле **SheetName**, значение **"B1"** в поле **Range** и текст **"Оценки>5"** в поле **Value**.
12. Добавить действие **Write Cell**. На панели свойств ввести имя **"Лист2"** в поле **SheetName**, значение **"B2"** в поле **Range** и **passedStud.ToString** в поле **Value**.
13. Запустить робот на выполнение. Проверить содержимое файла Excel.

Задание 3-3. Прочитать текст из документа, подготовленного в приложении Word, записать его в другой документ Word, добавить картинку и преобразовать в файл Pdf.

Пусть имеется файл с именем, например, **TextWord.docx**, содержащий документ Word с некоторым текстом и картинка с именем, например, **ImageRpa.jpg**.

1. Создать новый процесс и назвать его **WordToPdf**.
2. Поместить файлы **TextWord.docx** и **ImageRpa.jpg** в папку **WordToPdf**.
3. Перетащить последовательность **Sequence** на рабочую панель. Создать переменные:

Имя переменной	Тип переменной	Значение по умолчанию
sourceDocument	String	"TextWord.docx"
destinationDocument	String	"About Rpa.docx"
rawTextFromSource	String	

4. Добавить действие **Word Application Scope**.

Записать переменную **sourceDocument** в поле **File path**.

5. Добавить действие **Read Text** внутри блока **Сделать** действия **Word Application Scope**.

Записать переменную **rawTextFromSource** в поле **Text**.

6. Добавить действие **Word Application Scope** ниже предыдущего действия **Word Application Scope**.

Записать переменную **destinationDocument** в поле **File path**.

7. Щелкнуть по контейнеру **Сделать** и создать переменные:

Имя переменной	Тип переменной	Значение по умолчанию
picturePath	String	"ImageRpa.jpg"
exportedDocToPdf	String	"About Rpa.pdf"

8. Добавить действие **Append Text** в блок **Сделать**.

Записать переменную **rawTextFromSource** в поле **Text**.

9. Добавить действие **Add Picture** под действием **Append Text**.

Записать переменную **picturePath** в поле **PicturePath**.

10. Добавить действие **Export To PDF** под действием **Add Picture**.

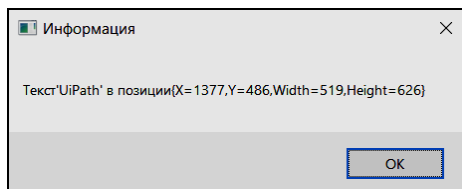
Записать переменную **exportedDocToPdf** в поле **File path**.

11. Запустить процесс на выполнение. Робот считывает текст из одного документа Word, записывает его в другой документ, добавляет картинку и преобразует документ из Word в PDF.

Лабораторная работа № 4. Извлечение данных из текста и изображений

Задание 4-1. Создать робот, который осуществляет ввод любой информации в приложение Блокнот и ее вывод с определением координат расположения текста.

1. Создать новый процесс с именем **FullTxt**. Открыть приложение Блокнот.
2. Перетащить контейнер **Sequence** на рабочую панель.
Создать переменную **myText** типа **String**.
3. Добавить действие **Input Dialog**.
На панели свойств ввести текст "Введите текст" в поле **Label**, текст "Ввод" в поле **Title**, переменную **myText** в поле **Результат**.
4. Добавить действие **Open Application**.
В окне действия нажать **Указать на экране**, щелкнуть по приложению Блокнот.
5. Поместить **Sequence** внутрь блока **Сделать (Do)** действия **Open Application**.
Создать переменные **fullText** типа **String**, переменную **position** типа **System.Drawing.Rectangle**.
6. Добавить действие **Type Into** внутрь **Sequence**.
В окне действия нажать **Указать на экране**, щелкнуть по приложению Блокнот.
На панели свойств ввести текст **myText.ToString** в поле **Text**.
7. Добавить действие **Get Full Text**.
В окне действия нажать **Указать на экране**, щелкнуть по приложению Блокнот.
На панели свойств ввести переменную **fullText** в поле **Text**.
8. Добавить действие **Get Position**.
В окне действия нажать **Указать на экране**, щелкнуть по приложению Блокнот.
На панели свойств ввести переменную **position** в поле **Rectangle**.
9. Добавить действие **Message Box**.
На панели свойств выбрать **Ok** в раскрывающемся списке **Buttons**, ввести текст "Информация" в поле **Caption**, выражение **"Текст '" + fullText.ToString + "'" в позиции '" + position.ToString** в поле **Text**.
10. Запустить процесс на выполнение.
В диалоговом окне надо ввести текст (например, UiPath), который автоматически записывается в Блокнот и затем обрабатывается. В конце отображается окно сообщения с информацией:



Задание 4-2. Создать робот в UiPath Studio, который прочитает отсканированное изображение счета и сохранит извлеченные данные в файле **.csv**

1. Открыть UiPath Studio, создать процесс и назвать его **ImageCsv**.
2. Перетащить действие **Sequence** из панели **Действия** и поместить его в рабочий процесс.

3. Добавить действие **Send Hotkey**, назвать его '**Открыть окно нажатием клавиш win + r**'. Выбрать **checkbox** (флажок) **Win** и ввести **r** в поле **Key (Клавиша)**.

4. Добавить действие **Type Into**.

Указать в кавычках путь к файлу изображения в поле **Text**.

Нажав клавиши **Win+r** (клавиша **Win** с изображением окон находится в левом нижнем углу клавиатуры рядом с клавишей **Ctrl**) открыть окно **Выполнить**, выбрать ссылку **Указать на экране** и щелкнуть на поле **Открыть окна Выполнить**.

5. Добавить действие **Send Hotkey**, назвать его '**Нажать кнопку ОК, чтобы открыть изображение**'. Ввести **enter** в поле **Key**.

6. Добавить действие **Get OCR Text**.

Нажав клавиши **Win+r** открыть окно **Выполнить**. Нажать на ссылку **Указать на экране**, нажать на клавишу **enter** и выделить область на изображении, из которой должны быть извлечены данные.

Создать переменную **InvoiceTotal** типа **String** и ввести имя в поле **Text**.

7. Добавить действие **Get OCR Text**.

Нажав клавиши **Win+r** открыть окно **Выполнить**. Нажать на ссылку **Indicate on screen (Указать на экране)**, нажать на клавишу **enter** и выделить область на изображении, из которой должны быть извлечены данные.

Создать переменную **EmailAddress** типа **String** и ввести имя в поле **Text**.

8. Добавить действие **Build Data Table**.

Создать переменную **ResultDT** типа **DataTable** и ввести имя в поле **DataTable**.

Сделать двойной щелчок по кнопке **DataTable** действия **Build Data Table**. В открывшемся окне дать названия столбцам: **InvoiceTotal**, **Email** типа **String**.

9. Добавить действие **Add Data Row**.

В свойстве **ArrayRow** указать массив переменных, из которых будут взяты данные для заполнения **Data Table**. Массив указывается в фигурных скобках через запятую, переменные в массиве должны идти в порядке их записи в соответствующие столбцы таблицы. В данном случае **{InvoiceTotal, EmailAddress}**.

Указать переменную **ResultDT** в свойстве **DataTable**.

10. Добавить действие **Write CSV**.

Указать переменную **ResultDT** в свойстве **DataTable**. Указать в свойстве **FilePath** путь и название файла, в который будет сохранен результат.

11. Запустить робот на выполнение.

Лабораторная работа № 5. Создание роботов, взаимодействующих с веб-сайтом

Задание 5-1. Создать робот в **UiPath Studio**, который будет извлекать информацию о сотрудниках с веб-сайта и сохранять их в файле **.xlsx**.

Пусть нужно извлечь информацию о сотрудниках с сайта генератора имен **https://www.fakenamegenerator.com**

1. Открыть **UiPath Studio**, создать процесс и назвать его **FromWebToExcel**.
2. Добавить действие **Sequence** из панели **Activities (Действия)** в окно рабочего процесса.

Создать переменные **Number**, **EmailToSend**, **URL** типа **String** значение по умолчанию **“https://www.fakenamegenerator.com”** и переменную **ExtractDataTable** типа **DataTable**.

3. Добавить действие **Input Dialog**.

В свойстве **Label** ввести текст **"Количество"**, в свойстве **Title** ввести текст **“Введите количество сотрудников”**, в свойстве **Result** ввести переменную **Number**.

4. Добавить действие **Build Data Table**.

Указать переменную **ExtractDataTable** в свойстве **DataTable**.

Сделать двойной щелчок по кнопке **DataTable** действия **Build Data Table**. В открывшемся окне создать один столбец с именем **Full Name Name** типа **String** в таблице.

5. Добавить действие **Open Browser**.

Указать переменную **URL** в свойстве **Url**.

6. Добавить действие **Sequence** в блок **Do** и назвать его **Sequence** – ‘Генерация фамилий сотрудников и запись их в таблицу’.

7. Добавить действие **Do While**.

Создать переменную **Val** типа **Int32** значение по умолчанию **0**, и в поле **Condition** указать условие **Val<cint(Number)**.

8. Добавить действие **Sequence** в блок **Body**.

9. Добавить действие **Assign** в последовательность.

Указать **Val** в поле **To**, а в поле **Value** ввести значение **Val+1**.

10. Добавить действие **Get Full Text**.

Нажать ссылку **Указать на...** и указать область в браузере, из которой необходимо извлечь информацию о фамилии сотрудника.

В свойстве **Text** указать переменную **FullName**.

11. Добавить действие **Add Data Row**.

В свойстве **Array Row** указать массив переменных, из которых будут взяты данные для заполнения **DataTable** в фигурных скобках через запятую. В данном случае **{FullName}**.

Указать переменную **ExtractDataTable** в свойстве **DataTable**.

12. Добавить действие **Click**.

Нажать ссылку **Указать на...** и указать кнопку на странице в браузере, на которую необходимо нажать.

13. Добавить действие **Excel Application Scope** после области действия **Open Browser**.

В текстовом поле действия указать имя файла Excel, который необходимо использовать в коде, например, **ExtrData.xlsx**.

14. Добавить действие **Write Range**.

В свойстве **Sheet Name** ввести текст **"Лист1"**, в свойстве **Range** ввести текст **"A1"**, в свойстве **Value** ввести переменную **ExtractDataTable**.

15. Запустить робот на выполнение.

Результат – файл ExtrData.xlsx:

G18			
	A	B	
1	Kristina C. Crockett		
2	Felicia E. Phelps		
3	Richard M. Holland		

Задание 5-2. Создать робот в UiPath Studio для заполнения веб-формы данными, извлеченными из таблицы Excel.

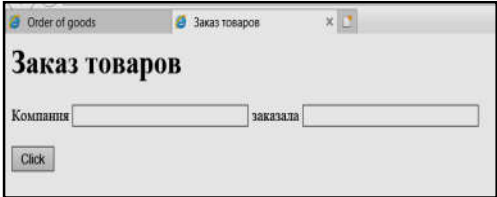
Пусть имеется файл Excel “**Firms.xlsx**” с информацией о компаниях и заказанных товарах.

Рассмотрим реализацию части задачи, которая заключается в заполнении двух текстовых полей на странице сайта содержимым двух ячеек таблицы Excel (вторая строка таблицы).

	A	B	
1	Фирмы	Товары	
2	Надежные системы	Ноутбуки	
3			

Чтобы создать простую страницу сайта можно в приложении Блокнот в файле с именем, например, **Zakaz.htm** записать текст, представленный ниже:

Текст файла Zakaz.htm	Вид страницы с формами
<pre><html> <head> <title>Заказ товаров</title> </head> <body><form name = "f1"> <h1 align = "left"> Заказ товаров </h1> Компания <input type = "text" name = "firma" size = 30> заказала <input type = "text" name = "tovar" size = 30>

 <input type = "button" name = "b1" size = 30 value="Click"> </form></body> </html></pre>	

Чтобы решить задачу в полном объеме надо написать программу для кнопки **Click**, которая будет обрабатывать полученные данные из текстовых полей. Тогда можно дополнить информацией файл Excel.

1. Открыть UiPath Studio, создать процесс и назвать его **FromTableToWeb**. Поместить в папку **FromTableToWeb** файлы **Zakaz.htm** и **Firms.xlsx**.

2. Перетащить действие **Sequence** из панели **Действия** в окно рабочего процесса.

3. Добавить действие **Open Browser**.

Указать в свойстве **Url** действия **Open Browser** ссылку на форму ("**D:\...\FromTableToWeb\Zakaz.htm**").

4. Добавить действие **Maximize Window** в последовательность **Do**.

5. Добавить действие **Excel Application Scope**.

В свойство **WorkbookPath** записать имя таблицы Excel ("**Firms.xlsx**"), из которой будет извлекаться информация. В таблице в ячейке **A1** должно находиться имя столбца **Company Name**, а в ячейке **B1** - **Goods**.

6. Добавить действие **Read Range**.

В свойстве **SheetName** указать имя листа ("**Лист1**"), а в свойстве **Range** указать "" (двойные кавычки).

Создать переменную **DtTable** типа **DataTable** и указать ее в свойстве **DataTable**.

7. Добавить действие **For Each Row**.

Указать переменную **DtTable** в поле **in (в)** и **row** в поле **ForEach**.

8. Добавить действие **Sequence** в раздел **Body**

Создать переменные **Company** и **Goods** типа **GenericValue**.

9. Добавить действие **Get Row Item** в контейнер **Sequence**.

Указать переменную **Company** в свойстве **Value (Значение)**, ввести текст "**Фирмы**" в свойстве **ColumnName**, в поле **Row** ввести **row**.

10. Добавить другое действие **Get Row Item**.

Указать переменную **Goods** в свойстве **Значение**, ввести текст "**Товары**" в свойстве **ColumnName**, в поле **Row** ввести **row**.

11. Добавить действие **Type Into**.

В свойствах указать переменную **Company** в поле **Text**.

Сделав двойной щелчок по файлу **Zakaz.htm** открыть в браузере Internet Explorer страницу с формами.

Нажать ссылку **Указать на...** и выделить на странице сайта первое текстовое поле.

12. Добавить другое действие **Type Into**.

В свойствах указать переменную **Goods** в поле **Text**.

Нажать ссылку **Указать на...** и выделить на странице сайта второе текстовое поле.

13. Добавить действие **Click**.

Нажать ссылку **Указать на...** и указать кнопку **Click** на странице в браузере.

14. Запустить робот на выполнение.

Лабораторная работа № 6. Проверка данных с использованием механизма Try-Catch и публикация проекта

Задание 6-1. Создать робот, который построит таблицу данных, и заполнит этими данными файл **.csv**. Для проверки введенных данных использовать механизм **Try-Catch**.

1. Создать процесс и назвать его **SimpleTryCatch**.
2. Перенести действие **Sequence** из панели **Действия** в рабочий процесс.
3. Добавить действие **Try Catch**.
4. Добавить действие **Sequence** внутрь блока **Try** и назвать 'Получение входных данных и ввод их в таблицу'.
Создать переменные **Name** и **Age** типа **String**, переменную **DtTable** типа **DataTable**.
5. Добавить действие **Input Dialog**.
Указать в свойствах в поле **Title** значение **"Имя"**, в поле **Label** текст **"Введите имя"**, в поле **Результат** переменную **Name**.
6. Добавить другое действие **Input Dialog**.
Указать в свойстве **Title** значение **"Возраст"**, в свойстве **Label** текст **"Введите возраст в правильном или неправильном формате"**, в поле **Результат** переменную **Age**.
7. Добавить действие **Build Data Table**.
В свойствах указать переменную **DtTable** в поле **Результат**.
Сделать двойной щелчок по кнопке **DataTable**. В открывшемся окне дать столбцам названия **Name** и **Age** (тип **String**).
8. Добавить действие **Add Data Row**.
В свойстве **ArrayRow** указать массив переменных, из которых будут взяты данные для заполнения таблицы: **{Name, Age}**.
Указать переменную **DtTable** в поле **DataTable**.
9. Добавить действие **Write CSV**.
Объявить переменную **DtTable** в поле **DataTable** и ввести название файла **"Untitled.csv"**, в который будут сохранены данные, в свойство **FilePath**.
10. Щелкнуть в блоке **Catches** и в раскрывающемся списке поля исключения выбрать **System.Exception**.
11. Добавить действие **Message Box** в созданную последовательность **Sequence**.
В свойствах в поле **Text** написать **"Исключение: добавлены недопустимые данные"**.
12. Запустить робот на выполнение.
Опробовать выполнение проекта при вводе данных в правильном и неправильном формате.
13. Изменить проект с тем, чтобы вводимые данные записывались в таблицу Excel.

Задание 6-2. Опубликовать проект из лабораторной работы № 1.

1. Открыть в UiPath Studio проект из лабораторной работы №1.
2. В нижней части экрана слева открыть панель **Проект**. В контекстном меню одного из файлов с расширением **.xaml** выполнить команду **Установить в качестве основного**.
3. В верхней части экрана перейти на вкладку **Проект** и нажать на кнопку **Публикация**. В появившемся окне выбрать радиокнопку **Параметры робота по умолчанию**.

В разделе **Примечания к выпуску** можно привести краткое описание проекта, установить флажок **Предварительный выпуск**.

Нажать **Опубликовать**. Если проект успешно опубликован, отобразится диалоговое окно **Info**.

4. Используя кнопку **Пуск**, открыть UiPath Robot.

UiPath Robot открывается в **фоновом** режиме. Для того, чтобы взаимодействовать с пользовательским интерфейсом робота, на панели задач в области уведомлений (в нижней части экрана справа) нужно найти иконку с буквами **Ui** и нажать на нее.

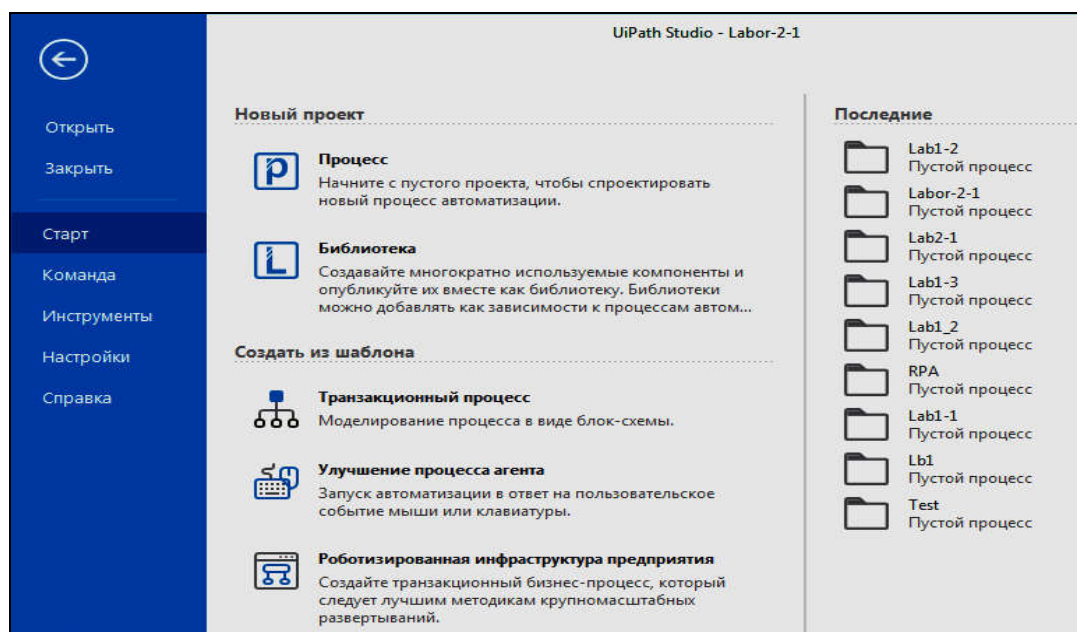
5. Открывается агент робота. Здесь можно увидеть опубликованный проект.

Если навести на него курсор, то справа от названия появится кнопка **Запуск**. Нажать на нее для запуска рабочего процесса.



Интерфейс UiPath Studio

Стартовое окно UiPath Studio:



1. **Процесс (Process)** используется для разработки простых роботов. Представляет собой чистый лист, где можно разрабатывать свои проекты с нуля.

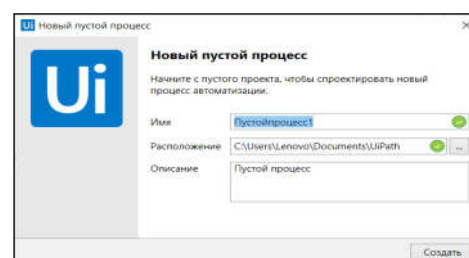
2. **Библиотека (Library)** позволяет создавать повторно используемые компоненты и публиковать их вместе в виде библиотеки.

3. **Транзакционный процесс (Transactional Process)** предоставляет шаблон блок-схемы, то есть диаграмму последовательности действий. Такие шаблоны используются предприятиями для создания роботов уровня предприятия, которые могут выполнять такие задачи, как ведение журнала, обработка ошибок и т. д. С помощью транзакционного процесса разработчик может смоделировать процесс как блок-схему.

4. **Улучшение процесса агента (Agent Process Improvement)** позволяет помочь любому пользователю, работающему в системе, автоматизировать задачу. Например, если пользователю нужно скопировать определенную задачу, то ему достаточно выбрать ее щелчком и нажать кнопку **Копировать**, либо использовать сочетание клавиш **Ctrl + C**.

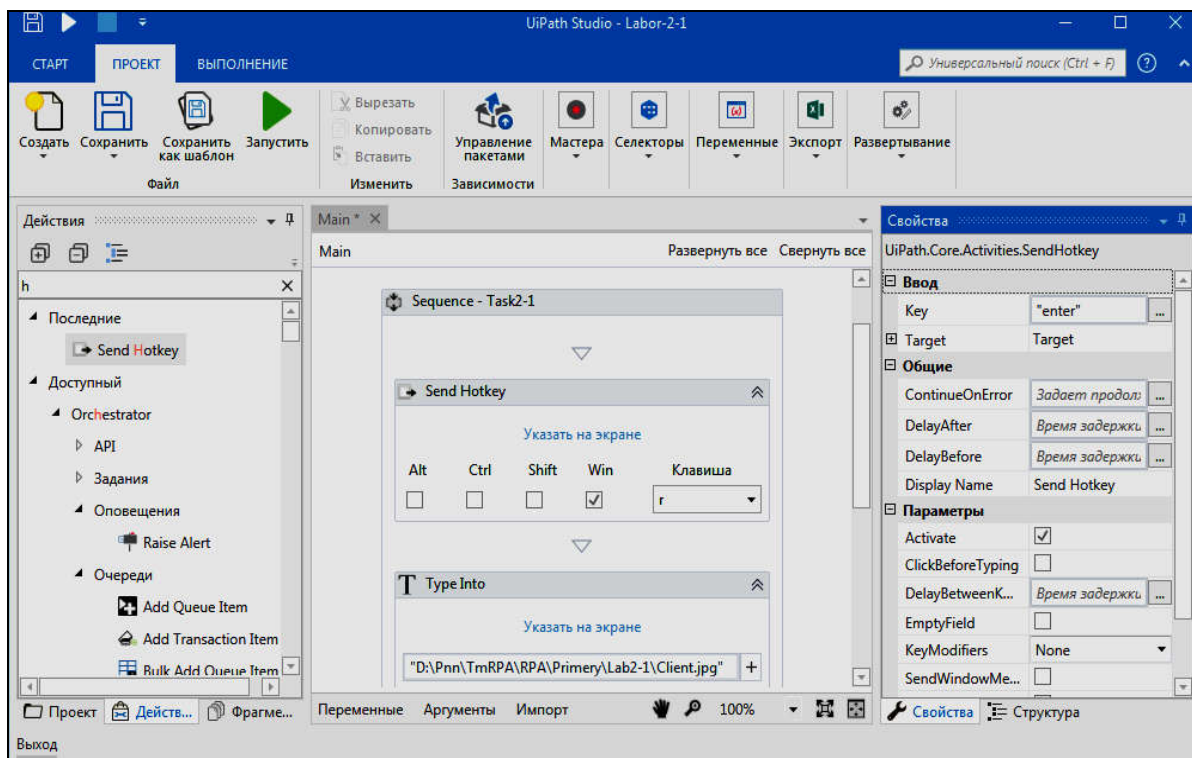
5. **Роботизированная инфраструктура предприятия (Robotic Enterprise Framework)** используется для определения состояний в проекте, которые полезны в бизнес-процессах. Когда пользователь имеет дело со сложными задачами, то важна последовательность выполнения действий. С помощью этого пункта обеспечивается контроль, при котором пользователь не может перейти на второй шаг без выполнения первого.

Создание проекта. Надо открыть UiPath Studio, выбрать **Процесс**, в появившемся окне ввести название проекта в поле **Имя**, определить расположение на диске в поле **Расположение** и краткое описание проекта в поле **Описание**. Нажать кнопку **Создать**.



Откроется главный экран UiPath Studio, в котором можно создавать роботов.

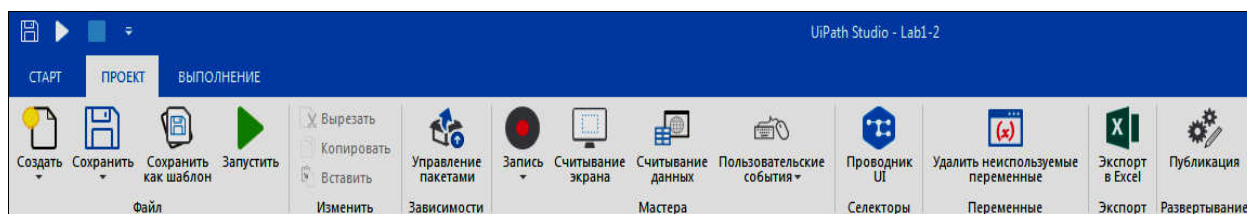
В центре экрана располагается рабочая **панель Дизайнер**. Это та область, куда перетаскиваются действия, и происходит построение роботов.



В самой **верхней** части экрана находится лента с тремя вкладками.

1. Вкладка **Старт (Start)** позволяет вернуться к стартовому окну UiPath Studio для того, чтобы открыть другой проект или создать новый, поменять настройки приложения, закрыть его и др.

2. На вкладке **Проект (Design)** располагаются основные инструменты, которые используются при разработке роботов.



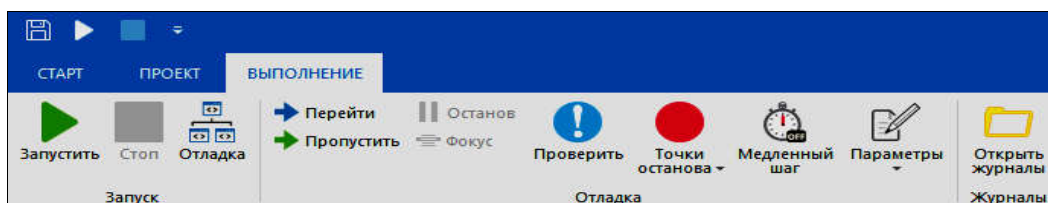
Основные компоненты вкладки:

- **New (Создать)** позволяет создать новый файл в проекте;
- **Save (Сохранить)** и **Save as Template (Сохранить как шаблон)** служат для сохранения файла с роботом;
- **Run (Запустить)** – запуск робота;
- **Manage Packages (Управление пакетами)** позволяет устанавливать пакеты и управлять ими;
- **Recording (Запись)** – запись движения мыши и действий клавиатуры для создания сценариев автоматизации;
- **Screen Scraping (Считывание экрана)** и **Data Scraping (Считывание данных)** позволяют извлекать информацию (неструктурированную и структурированную);
- **User Events (Пользовательские события)** – с помощью этого действия можно обрабатывать пользовательские события;
- **UI Explorer (Проводник UI)** позволяет открыть инструмент для создания пользовательского селектора для определенного элемента пользовательского интерфейса. Селекторы

являются фундаментальной частью **UiPath** для идентификации отдельных элементов интерфейса на экране;

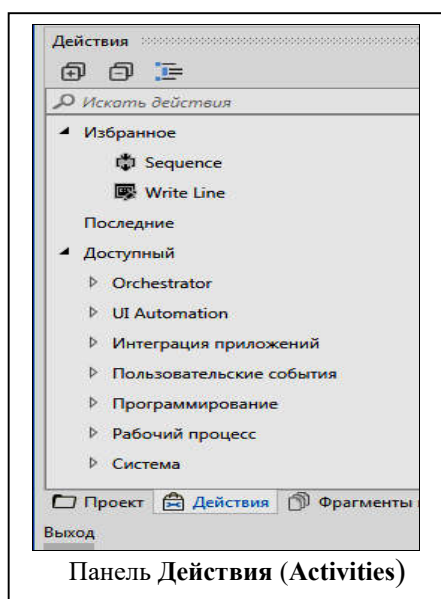
- **Remove Unused Variables (Удалить неиспользуемые переменные)** – удаление неиспользуемых переменных;
- **Export to Excel (Экспорт в Excel)** – экспорт рабочего процесса в **Excel**.
- **Publish (Публикация)** позволяет опубликовать робот в **Orchestrator**.

3. На вкладке **Выполнение (Execute)** располагаются инструменты, которые используются при отладке роботов.



Основные компоненты вкладки:

- **Run (Запустить)** – запуск робота;
- **Stop (Стоп)** – остановка выполнения робота и возврат к режиму разработки;
- **Debug (Отладка)** – запуск робота в режиме отладки;
- **Continue (Продолжить)** позволяет продолжить выполнение робота (появляется после запуска робота в режиме отладки);
- **Step Into (Перейти)** позволяет отлаживать действия по одному, например, если в процессе отладки встречается последовательность действий, то все действия выполняются по очереди;
- **Step Over (Пропустить)** – отладка действий по одному (может пригодиться для пропуска некоторых действий);
- **Break (Остановить)** позволяет сделать паузу в режиме отладки;
- **Focus (Фокус)** помогает вернуться к текущей точке останова или к действию, которое вызвало ошибку во время отладки;
- **Validate (Проверить)** – проверка рабочего процесса на наличие ошибок;
- **Breakpoints (Точки останова)** позволяет поставить на элементе точку останова (преднамеренное прерывание выполнения программы);



Панель Действия (Activities)

- **Slow Step (Медленный шаг)** служит для внимательного изучения действий во время отладки;
- **Options (Параметры)** позволяет настроить некоторые параметры отладки;
- **Open Logs (Открыть журналы)** – открытие папки, в которой хранятся файлы журналов.

В левой части экрана располагается **панель Действия (Activities)**. Она содержит основные действия, которые пользователь может выполнять на компьютере: щелчок, открытие, ввод, чтение. Здесь имеется строка **Искать действия (Search activities)** для поиска нужного действия. Например, если надо нажать на кнопку, то ищется слово **click** и, после его появления на экране, можно мышкой перетащить слово на панель дизайнера.

Панель содержит семь разделов:

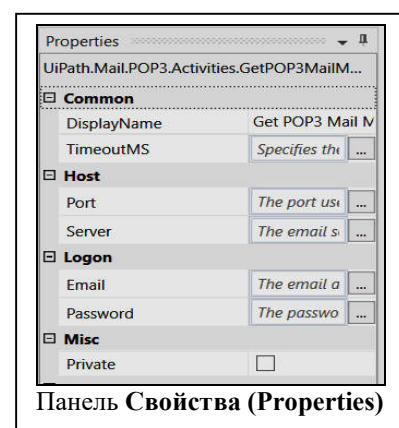
- **UI Automation (Автоматизация UI)** включает все элементы пользовательского интерфейса, например, наведение курсора мыши на что-нибудь, выделение, нажатие на изображение, извлечение текста и др.;

- **User Events (Пользовательские события)** содержит триггеры, которые срабатывают всякий раз, когда происходит какое-то событие, например, событие «копирование» при нажатии Ctrl+C;
- **App Integration (Интеграция приложений)** включает действия, связанные с интеграцией приложений, например, действия для работы с Excel (запись в файл, чтение из файла), для работы с почтой (получение и отправка электронных писем) и др.
- **Orchestrator (Оркестратор)** используется при работе с несколькими компьютерами при необходимости управления несколькими роботами;
- **System (Система)** – действия, которые используются при выполнении таких операций, как копирование в буфер обмена, создание файла, удаление файла, открытие командной строки и др.;
- **Programming (Программирование)** – действия, связанные с программированием, например, при работе с коллекциями, таблицами данных и др.;
- **Workflow (Рабочий процесс)** – действия, которые используются внутри блок-схем, например, циклы (while, do while, for each), try catch (для обработки ошибок), if (условная инструкция) и др.

В правой части экрана находится панель **Свойства (Properties)**, которая отображает все связанные с действием свойства.

Эти свойства можно менять в соответствии с требованиями.

Например, если перетащить какое-либо действие на панель дизайнера и щелкнуть по нему, то на панели справа отобразятся свойства этого действия.



В нижней части экрана располагаются различные **вкладки**.

Вкладка **Переменные (Variables)**, при раскрытии которой появляется соответствующая **панель**. С ее помощью создаются переменные и, при необходимости, в них помещаются нужные значения. Панель состоит из следующих полей:

- Name (Имя)** – обязательное поле, в котором записывается имя переменной;
- **Variable Type (Тип переменной)** – обязательное поле, в котором обозначается тип переменной;
- **Scope (Область)** – область, в которой доступна переменная, по умолчанию переменные доступны во всем проекте;
- **Default (По умолчанию)** – необязательное поле, в которое может записываться значение по умолчанию для переменной.

Name	Variable type	Scope	Default
variable1	String	Sequence	Enter a VB expression
variable2	Boolean	Sequence	Enter a VB expression
variable3	Int32	Sequence	34
Create Variable			

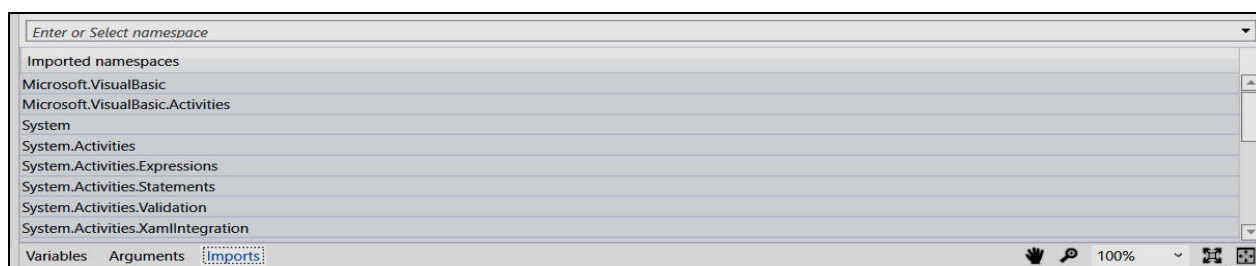
С помощью вкладки **Аргументы (Arguments)** создаются аргументы, которые используются, когда выполняется работа с несколькими файлами.

Name	Direction	Argument type	Default value
argument1	In	String	"Something"
argument2	Out	Int32	Default value not supported
argument3	In/Out	Boolean	Default value not supported
Create Argument			

Например, используется основной файл и нужно вызвать последовательность, определенную в другом файле. Тогда должны быть определены аргументы в основном файле для передачи их в другой файл. Панель состоит из следующих полей:

- **Name (Имя)** – обязательное поле, хранит название аргумента;
- **Direction (Направление)** – обязательное поле, хранит направление для аргумента. Доступны следующие опции: **In (Входной)** – может использоваться только внутри данного проекта, **Out (Выходной)** – может использоваться для передачи данных за пределы данного проекта, **In / Out (Входной/Выходной)** – может использоваться как внутри, так и за пределами проекта, **Property (Свойство)** – в настоящее время не используется;
- **Argument Type (Тип аргумента)** – обязательное поле, хранит тип значения, которое будет храниться в аргументе;
- **Default value (Значение по умолчанию)** – необязательное поле, хранит значение по умолчанию для аргумента.

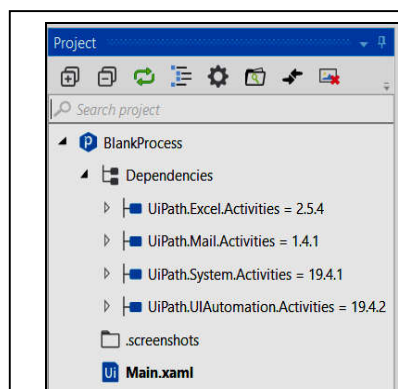
На вкладке **Импорт (Imports)** отображаются импортированные пространства имен.



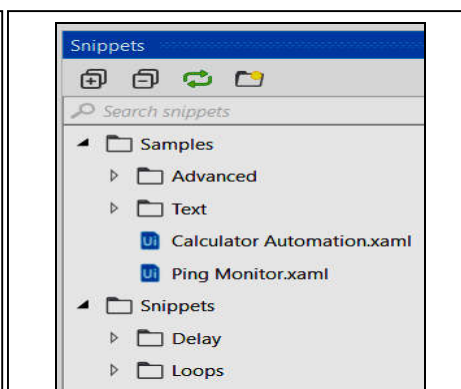
На вкладке **Проект (Project)** можно просматривать содержимое текущего проекта, добавлять папки, открывать файлы, управлять зависимостями и настраивать параметры проекта.

Вкладка **Фрагменты кода (Snippets)** позволяет повторно использовать код. По умолчанию она включает несколько примеров и фрагментов кода. Можно добавить свой собственный пример, нажав кнопку **Add Folder (Добавить папку)** и выбрав каталог на диске. Чтобы удалить папку, надо щелкнуть по ней правой кнопкой мыши и выбрать команду **Remove (Удалить)**.

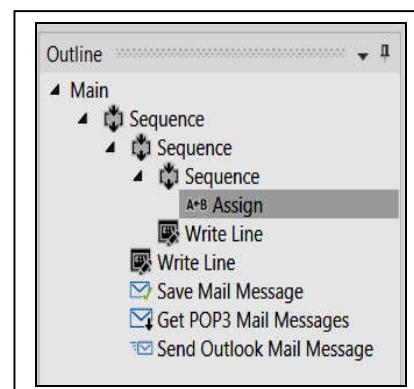
Вкладка **Структура (Outline)** отображает иерархию проекта, все доступные переменные и узлы. Можно выделить действия на этой панели, выбрав их на панели **Designer (Дизайнер)**, или перейти к определенной деятельности, выбрав ее на панели **Outline (Структура)**.



Панель **Проект (Project)**

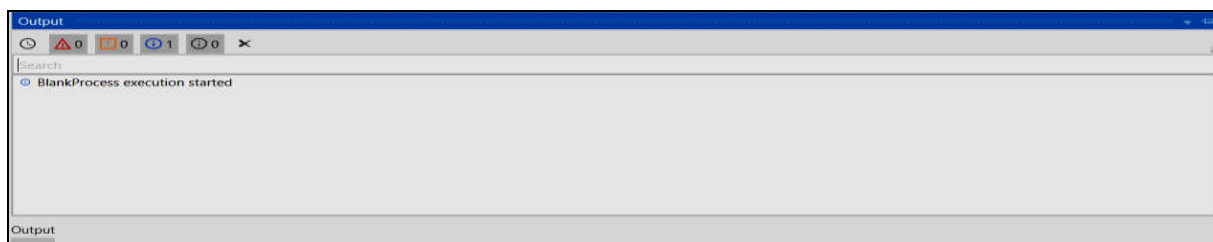


Панель **Фрагменты кода (Snippets)**



Панель **Структура (Outline)**

Вкладка **Выход (Output)** позволяет отображать результаты действий **Log Message** или **Write Line**. На панели **Output (Выход)** можно отображать или скрывать сообщения с разными уровнями журнала (ошибки, предупреждения), нажимая кнопки в заголовке панели. Двойной щелчок на сообщении отображает дополнительную информацию о нем, с возможностью копирования информации.



При отладке на этой панели можно отобразить журналы с помощью параметра **Options (Параметры)** => **Log Activity (Записывать действия в журнал на вкладке Debug (Отладка))**.