Лабораторная работа № 4. Обработчики ошибок. Использование

модулей и библиотек. Публикация проектов

**Задание 4-1**. Создать процесс автоматизации, который построит таблицу данных, и заполнит этими данными файл **.csv**. Для проверки введенных данных использовать механизм **Try-Catch**.

1-3. Создаем процесс, добавляем последовательность и в неё добавляем действие TryCatch(Рисунок 4.1)

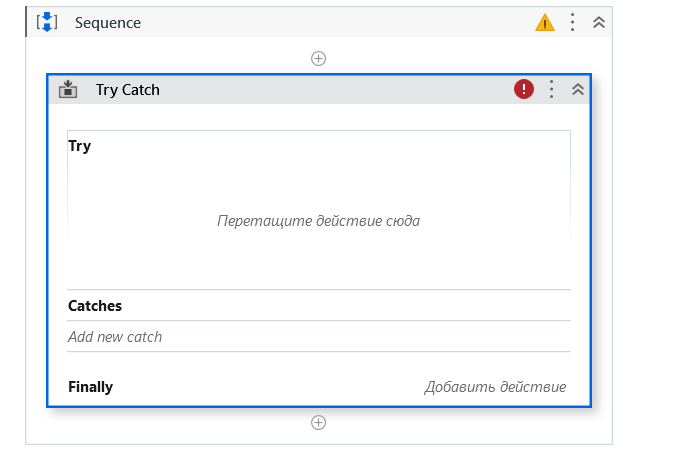


Рисунок 4.1­–Добавление действия TryCatch

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**. Представлено на рисунке 4.2.

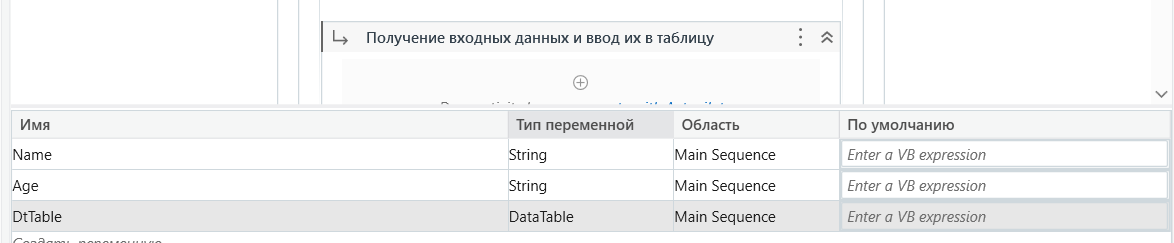


Рисунок 4.2 – Добавление Sequence и переменных

5. Добавить действие **Input Dialog**.

Указать в свойствах в поле **Title** значение **"Имя"**, в поле **Label** текст **"Введите имя**", в поле **Результат** переменную **Name**.

Результат на рисунке 4.3.

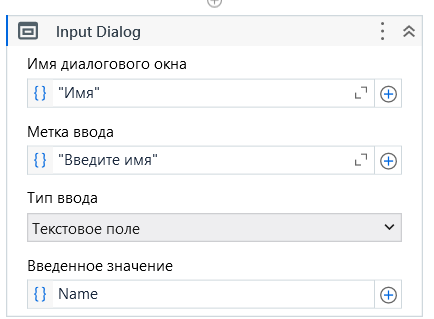


Рисунок 4.3 – Окно Input Dialog для поля «Имя»

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**.

Результат на рисунке 4.4.

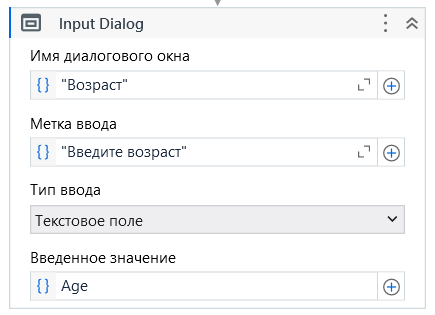


Рисунок 4.4 – Окно Input Dialog для поля «Возраст»

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** (тип **String**)и **Age** (тип **Int32**). Рисунок 4.5.

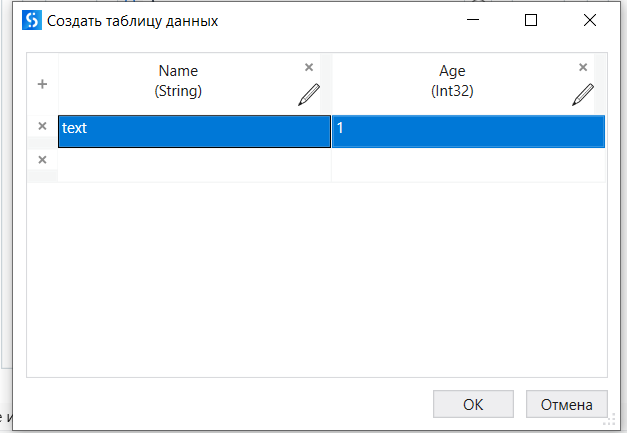


Рисунок 4.5 – Создание таблицы для данных

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**. Выполнение на рисунке 4.6.

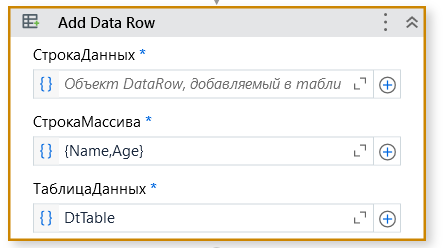


Рисунок 4.6 – Добавление действия Add Data Row

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**. Рисунок 4.7.

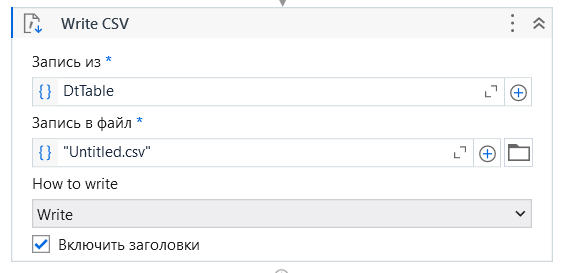


Рисунок 4.7 – Действие Write CSV

10. Щелкнуть в блоке **Catches** и в раскрывающемся списке поля исключения выбрать **System.Exception**. Рисунок 4.8.

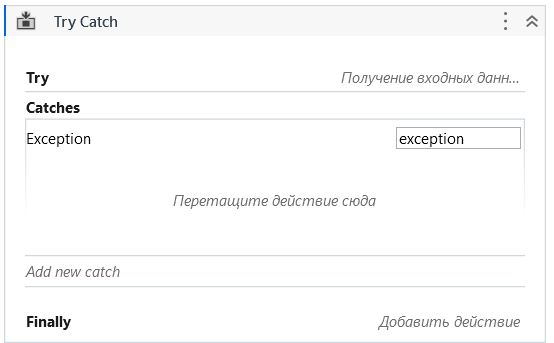


Рисунок 4.8 – Блок Catch

11. Добавить действие **Message Box** в созданную последовательность **Sequence.**

В свойствах в поле **Text** написать **"Исключение: добавлены недопустимые данные".** Рисунок 4.9.

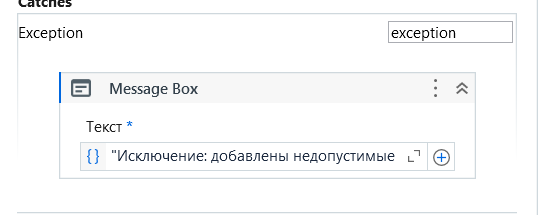


Рисунок 4.9 – Добавление Message Box

12.Запустить процесс автоматизации на выполнение. Опробовать выполнение проекта при вводе данных в правильном и неправильном формате. На рисунке 4.10 результат работы.

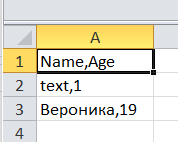
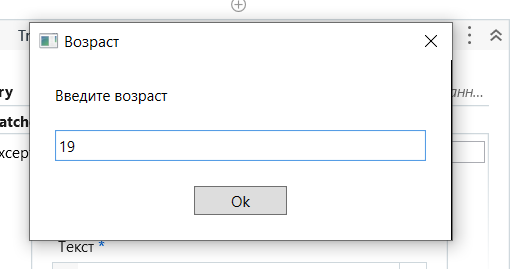
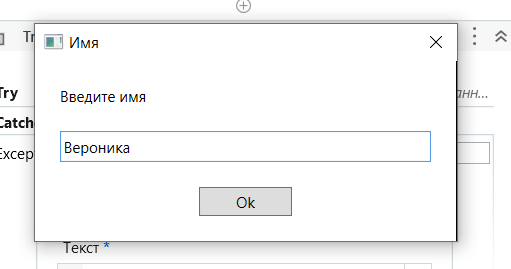


Рисунок 4.10 – Результат выполнения

**Задание 4-2**. Создать процесс автоматизации с использованием Глобального обработчика исключений, который после ввода текста в Блокнот закроет приложение.

Если документ сохранить нажатием клавиш **Ctrl-s** после автоматического ввода текста, то процесс заканчивается правильно. Если не сохранять документ, то появится сообщение об ошибке.

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

2. Открыть Блокнот и сохранить файл.

3. Добавить действие **[Type Into](https://activities.uipath.com/docs/type-into" \t "_self)** на рабочую панель. Действие будет автоматически добавлено в последовательность **Sequence**.

Ввести текст, например, “**Глобальный обработчик исключений**” в поле **Text** (этот текст будет записан в Блокнот). Рисунок 4.11

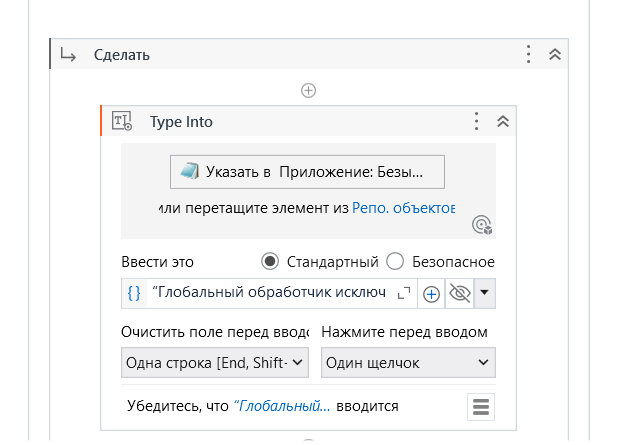


Рисунок 4.11 – Добавление Type Into

Нажать ссылку **Указать на экране**. Окно Studio сворачивается и позволяет щелкнуть внутри окна Блокнота.

4. Добавить действие **Send Hotkey** в рабочий процесс.

Отметить **Ctrl**, ввести в поле **Key** значение **s** (для сохранения документа Блокнота после ввода текста). Указать на окно Блокнота. Рисунок 4.12.

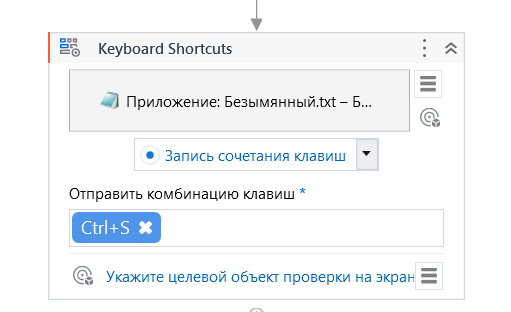


Рисунок 4.12 – Сохранение клавиш

5. Добавить в рабочий процесс действие **Close Application**.

Указать на окно Блокнота.

**Добавление в проект глобального обработчика исключений**

6. На **ленте** **Проект** **нажать кнопку Новая и выбрать Глобальный обработчик**.

Откроется окно, в котором надо ввести **Имя** обработчика и сохранить его в папке проекта. Рисунок 4.13.

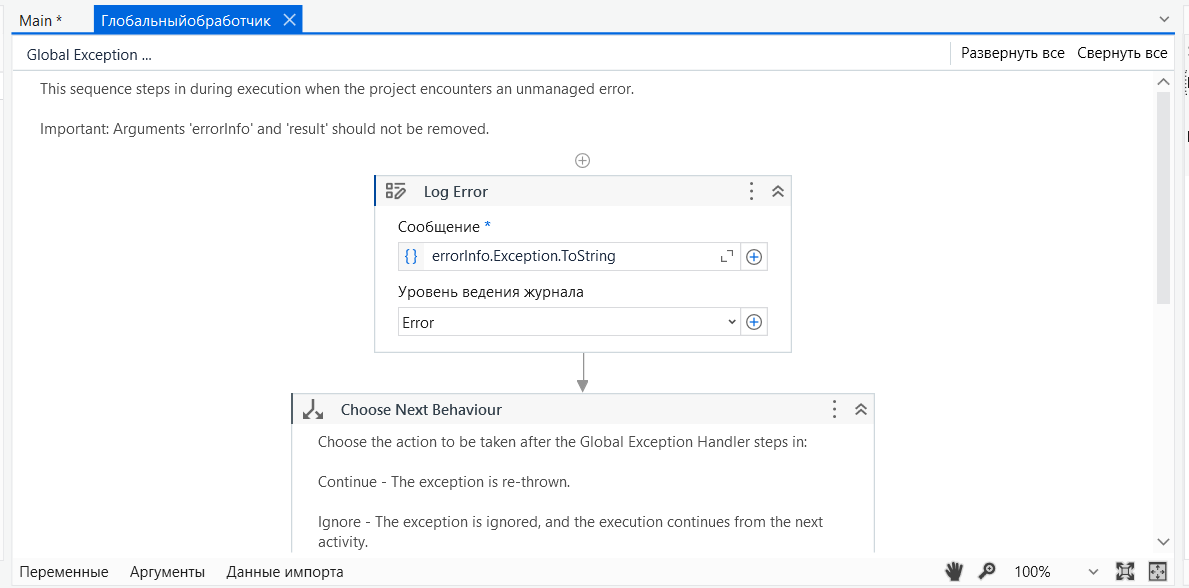


Рисунок 4.13 – Глобальный обработчик

7. Запустить процесс автоматизации **Main** на выполнение.

После записи текста в Блокнот нажать вместе клавиши **Ctrl-s**, чтобы правильно закрыть и сохранить файл. Результат на рисунке 4.14.

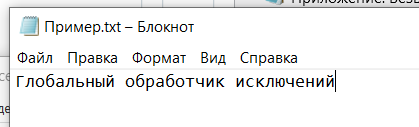
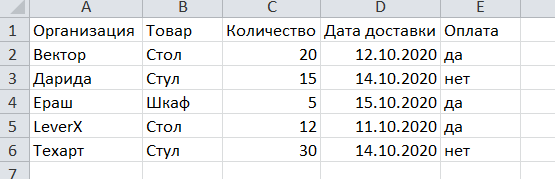


Рисунок 4.14 – Результат программы

**Задание 4-3**. Создать процесс автоматизации, демонстрирующий способы использования **модуля** в процессе автоматизации с действием **Invoke Workflow File**. Процесс автоматизации должен анализировать содержимое таблицы Excel, в которой записана информация о заказанных организациями товарах и оплате за них. Если товар не оплачен, то название организации следует вывести в окно **Вывод**.

Пусть имеется таблица Excel c информацией о заказанных организациями товарах с именем **Заказы.xlsx**.



В **модуле** сравнивается содержимое столбца с признаком оплаты со словом "Нет". Если товар не оплачен, то в окно **Вывод** выводится соответствующее название организации.

В основной программе читается таблица и для каждой строки осуществляется обращение к модулю.

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

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

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

Ввести путь к таблице Excel "**Заказы.xlsx**", если таблица лежит в папке **OrderModule** или ввести полный путь к файлу, если таблица в другой папке, в свойство **Workbook path**.

5. Добавить действие **Read Range**.

В поле **SheetName** ввести текст **"Лист1"**, в поле **Range** ввести **"A3:E8"**, в поле **DataTable** создать переменную **DTable**, используя клавиши **Ctrl-K**.

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

Указать **row** в секции **For Each** и переменную **DTable** в поле **DataTable**.

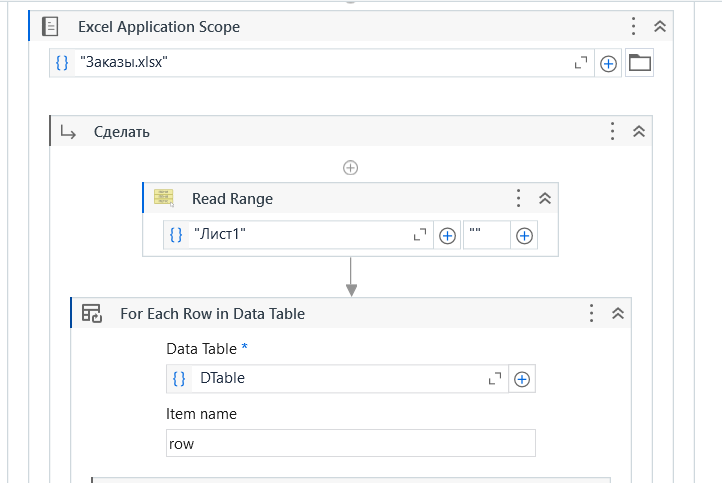


Рисунок 4.15 – Использование вышеперечисленных действий

7. На ленте **Проект** нажать кнопку **Новая**, выбрать пункт **Последовательность** ив окне указать имя нового модуля, например, **ModulAnaliz**.

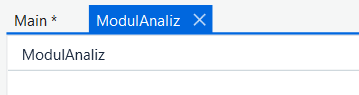


Рисунок 4.16 – Создание последовательности

На панели **Аргументы** создать входные аргументы **in\_Payment** и **in\_Company** типа **String**.

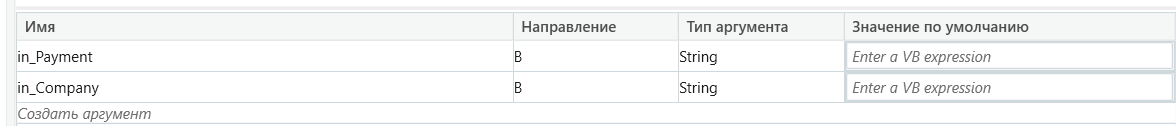


Рисунок 4.17 – Создание аргументов

8. Вернуться в проект **Main** и добавить в разделе **Body** действие **Invoke Workflow File**.

Внутри действия нажать кнопку с изображением папки и выбрать ранее созданную последовательность **ModulAnaliz**.

Нажать **Импортировать аргументы**. Откроется окно **Аргументы вызываемого рабочего процесса (Invoked Workflow’s Arguments)**, где отображаются аргументы первой последовательности.

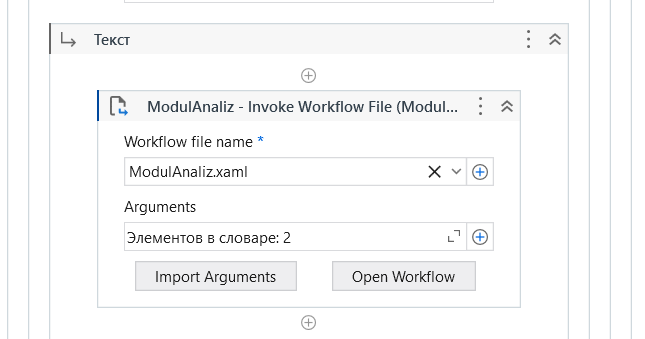


Рисунок 4.18 – Действие Invoke Workflow File

В поле **Имя** ввести **in\_Payment** со значением **row("Оплата").ToString** и во втором поле **Имя** − **in\_Company** со значением **row("Организация").ToString**.

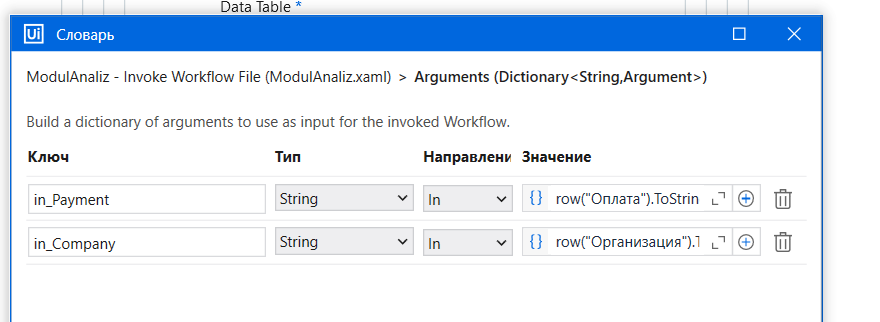


Рисунок 4.19 – Импорт аргументов

Нажать **OK**. Аргументы импортируются.

9. Перейти к модулю **ModulAnaliz**.

Добавить действие [**If**](https://activities.uipath.com/docs/message-box).

В поле **Condition** записать выражение **in\_Payment ="Нет"**.

10. В блок **Then** добавить действие **Write Line**.

В поле **Text** ввести аргумент **in\_Company**.

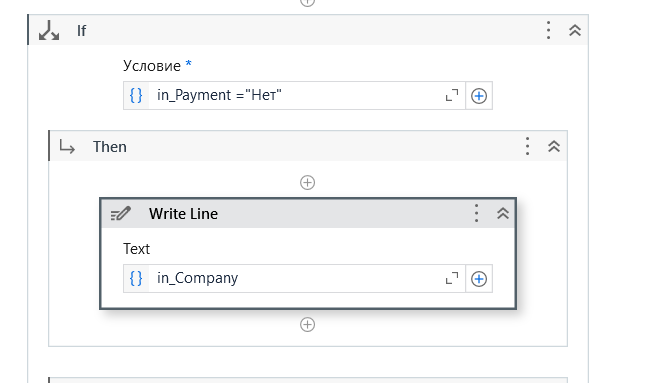


Рисунок 4.20 – Содержимое последовательности

11. Вернуться в проект **Main** и запустить процесс автоматизации на выполнение. Проверить содержимое окна **Вывод**.

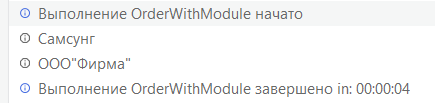


Рисунок 4.21 – Вывод данных

**Задание 4-4**. Создать проект, в котором данные получаются из одной таблицы Excel и добавляются в другой файл Excel. Затем опубликовать проект в библиотеке в особом канале, установить в другом проекте и использовать содержащиеся в нём действия.

1. Открыть UiPath Studio и выбрать пункт **Библиотека**, чтобы создать новый проект в библиотеке.

В появившемся окне **Новая пустая библиотека в**вести имя проекта, например, **LibraryFirst**. Добавить краткое описание, выбрать место расположения и нажать **Создать**.

Новая библиотека откроется в Studio на панели **Проект**. Нажать **NewActivity.xaml**.

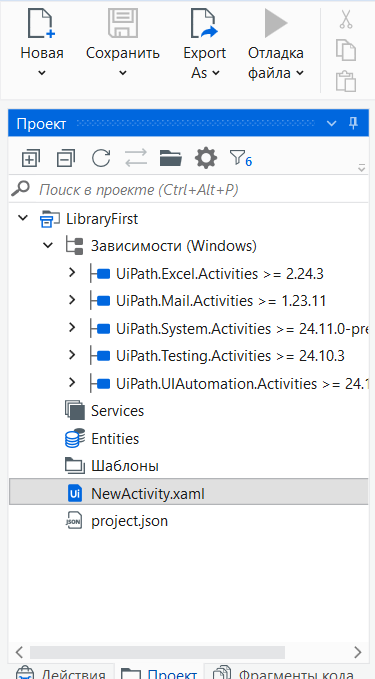


Рисунок 4.22 – NewActivity.xaml

2. На панели **Аргументы** создать входной аргумент **in\_CurDir** типа **String** со значением по умолчанию **Directory.GetCurrentDirectory** (это необходимо для того, чтобы пути интерпретировались не относительно библиотеки, а относительно запускаемого проекта).

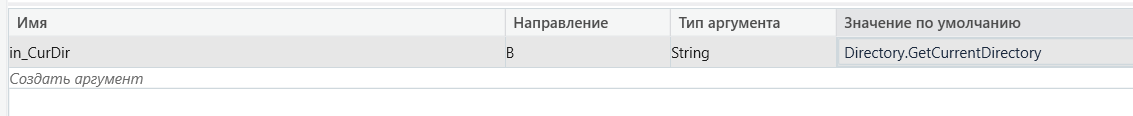


Рисунок 4.23 – Добавление аргументов

3. Добавить действие **Read Range из раздела Система**.

В поле **Путь к рабочему журналу** добавить путь к файлу Excel, из которого нужно получить данные (например, **Path.Combine(in\_CurDir,“Examen.xlsx")**), при этом файл должен находиться в папке **LibraryFirst**, в противном случае надо прописать полный путь к файлу).

Указать имя листа, например, “**Лист1**”, а поле **Range** оставить пустым (если поле пустое, то читаются все ячейки).

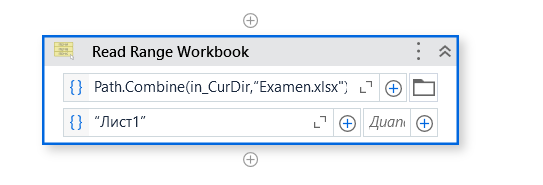


Рисунок 4.24 – действие Read Range Workbook

На панели свойств в поле **DataTable** при помощи сочетания клавиш **Ctrl+K создать** переменную **dtTable**. Автоматически будет создана переменная типа **DataTable.**

**В свойстве** **AddHeaders проверить наличие галочки. Если убрать галочку из свойства AddHeaders, то будет считываться текст вместе с заголовками.**

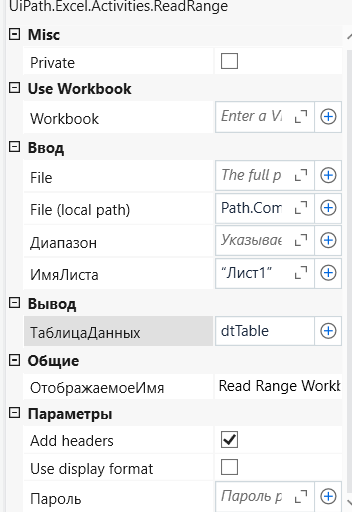


Рисунок 4.25 – Установка нужных значений в свойствах

4. Добавить действие **Append Range** **из раздела Система**.

В поле **Путь к рабочему журналу** добавить в кавычках путь к файлу Excel, в который надо добавить данные, например, **Path.Combine(in\_CurDir,“Examen2.xlsx")**.

Указать имя листа, например, “**Лист1**”.

В поле **DataTable** добавить ранее созданную переменную **dtTable**.

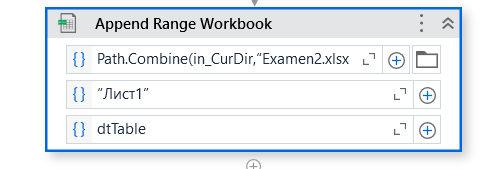


Рисунок 4.26 – действие Append Range Workbook

5. **Запустить** проект на выполнение.

Данные, собранные из первой таблицы Excel, добавляются в другую.

**Публикация библиотеки в особый канал**

Чтобы использовать этот проект в других проектах автоматизации в качестве многократно используемого компонента, нужно преобразовать его в пакет (файл .**nupkg)**.

1. Открыть выше созданный проект в Studio.

2. Нажать кнопку **Опубликовать** на ленте **Проект**.

В появившемся окне на вкладке **Свойства пакета** можно привести краткое описание изменений, внесенных в проект, добавить новую версию в поле **Новая версия** и установить флажок **Предварительная версия**, чтобы пометить версию как alpha, если это необходимо.

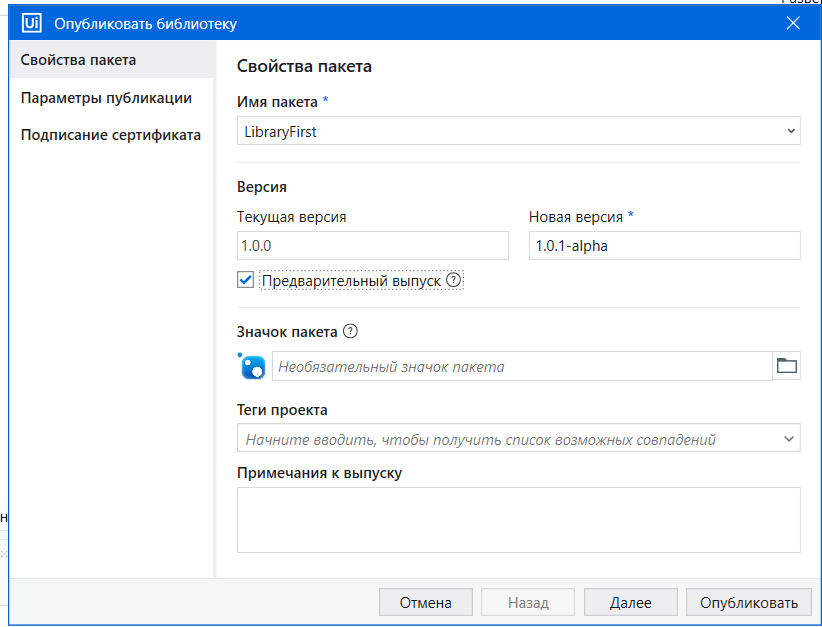


Рисунок 4.27 – Свойства публикации библиотеки

На вкладке **Параметры публикации** в поле **Особый URL-адрес** ввести адрес папки, где будут храниться пакеты .**nupkg**.

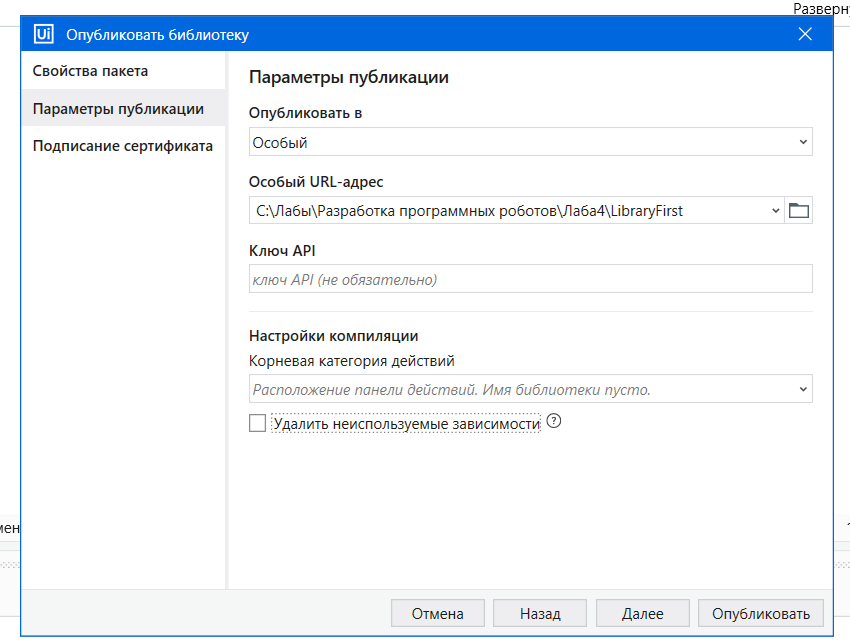


Рисунок 4.28 – Параметры публикации библиотеки

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

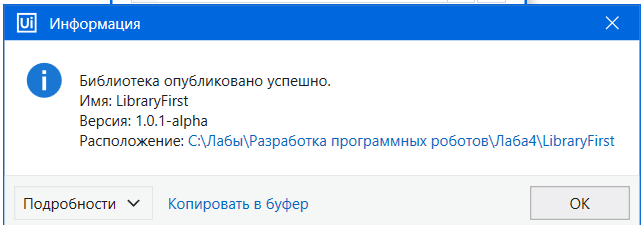


Рисунок 4.29 – Успешное выполнение

**Добавление пользовательского канала**

1. На ленте **Проект** нажать кнопку **Управление пакетами > Настройки**. Откроются каналы по умолчанию и каналы, определенные пользователем.

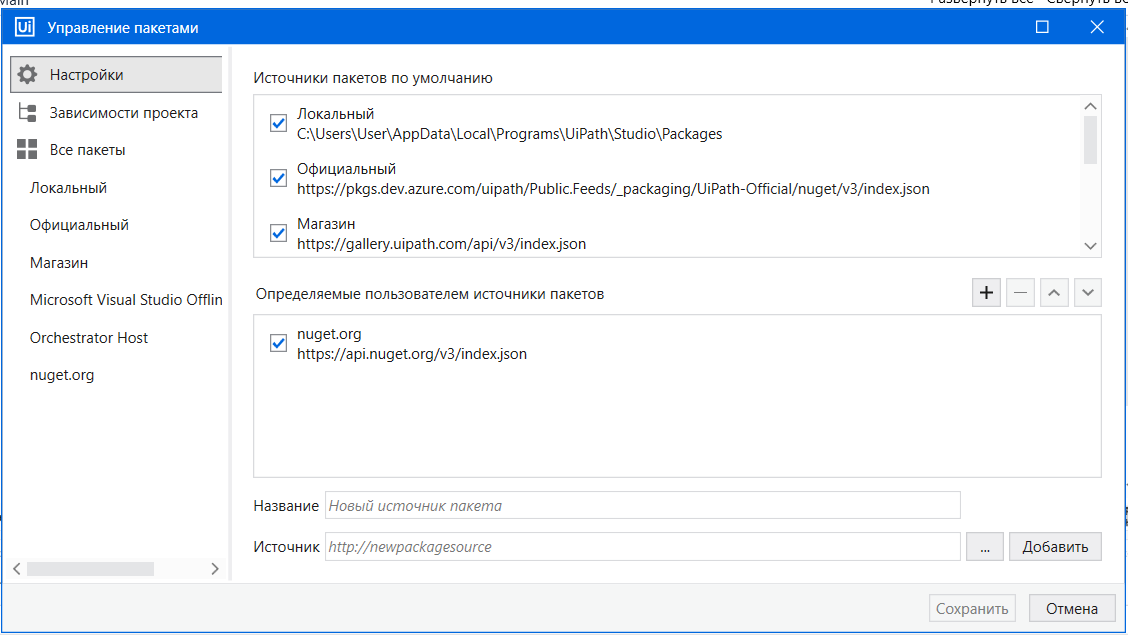


Рисунок 4.30 – Окно управления пакетами

2. В контейнере **Определенные пользователем источники пакетов** добавить имя своего канала (например, **myChannel**) в поле **Имя**.

В поле **Источник** добавить путь к папке, где опубликован проект библиотеки.

3. Нажать **Добавить**. Новый канал будет добавлен в раздел.

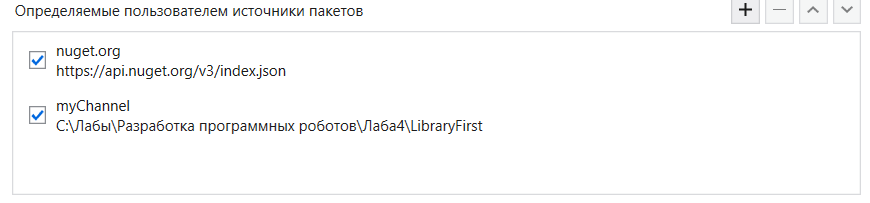


Рисунок 4.31 – Добавлен новый источник пакетов

**Установка пакета**

1. Создать новый процесс и назвать его **UsingTheLibrary.** Поместить в папку проекта файл **Examen.xlsx**

В окне **Управление пакетами** перейти к ранее добавленному каналу на вкладке **Все пакеты**, расположенной слева. Откроется список пакетов, доступных в канале.

2. Найти и выбрать пакет, в данном примере имя пакета **LibraryFirst**.

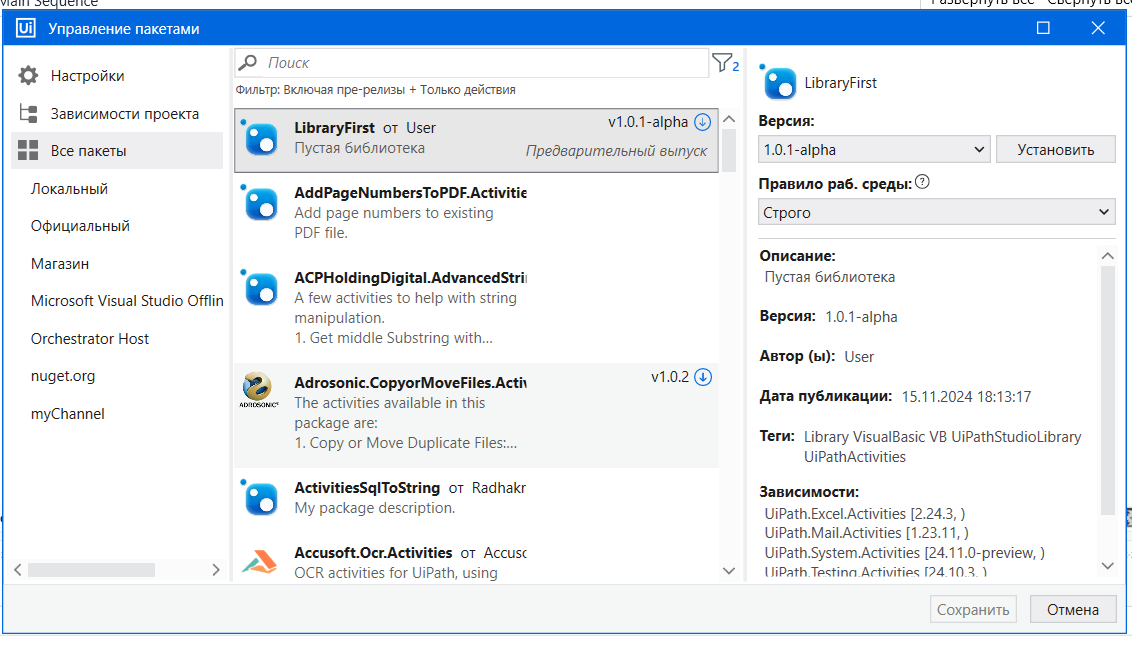


Рисунок 4.32 – Выбор нужного пакета

3. Нажать **Установить**, затем **Сохранить**. Теперь пакет установлен и виден на панели **Проект** среди **Зависимостей**.

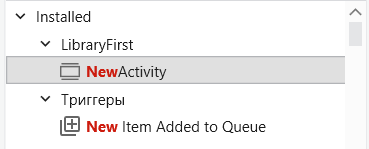


Рисунок 4.32 – Отображение в проекте

**Добавление действий из библиотеки**

1. На вкладке **Проект** найти имя пакета, в данном примере **LibraryFirst**. Здесь отображается древовидное представление с папкой проекта и зависимостями.

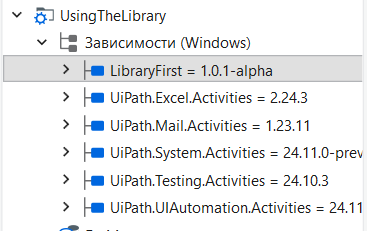


Рисунок 4.33 – Отображение древовидного представления

2. Перейти на вкладку **Действия** и переместить действие **NewActivity** на рабочую панель.

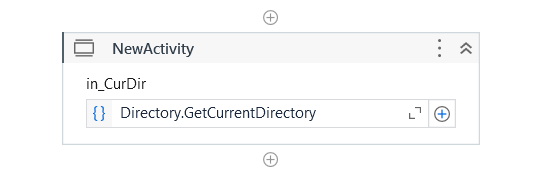


Рисунок 4.34 – Добавление действия NewActivity

**Задание 4-5**. Поместить другой проект из какой-либо лабораторной работы в библиотеку и опубликовать. Продемонстрировать использование получившегося проекта.

