**Лабораторная работа № 3. Работа с информацией из изображений, сайтов,** **электронной почты**

**Задание 3-1**. Создать процесс автоматизации в UiPath Studio, который прочитает отсканированное изображение счета и сохранит извлеченные данные в файле **.csv**

Для реализации процесса нам нужна картинка со счётом. Шаблон можно найти и заполнить на сайте в интернете и после сохранить в качестве картинки

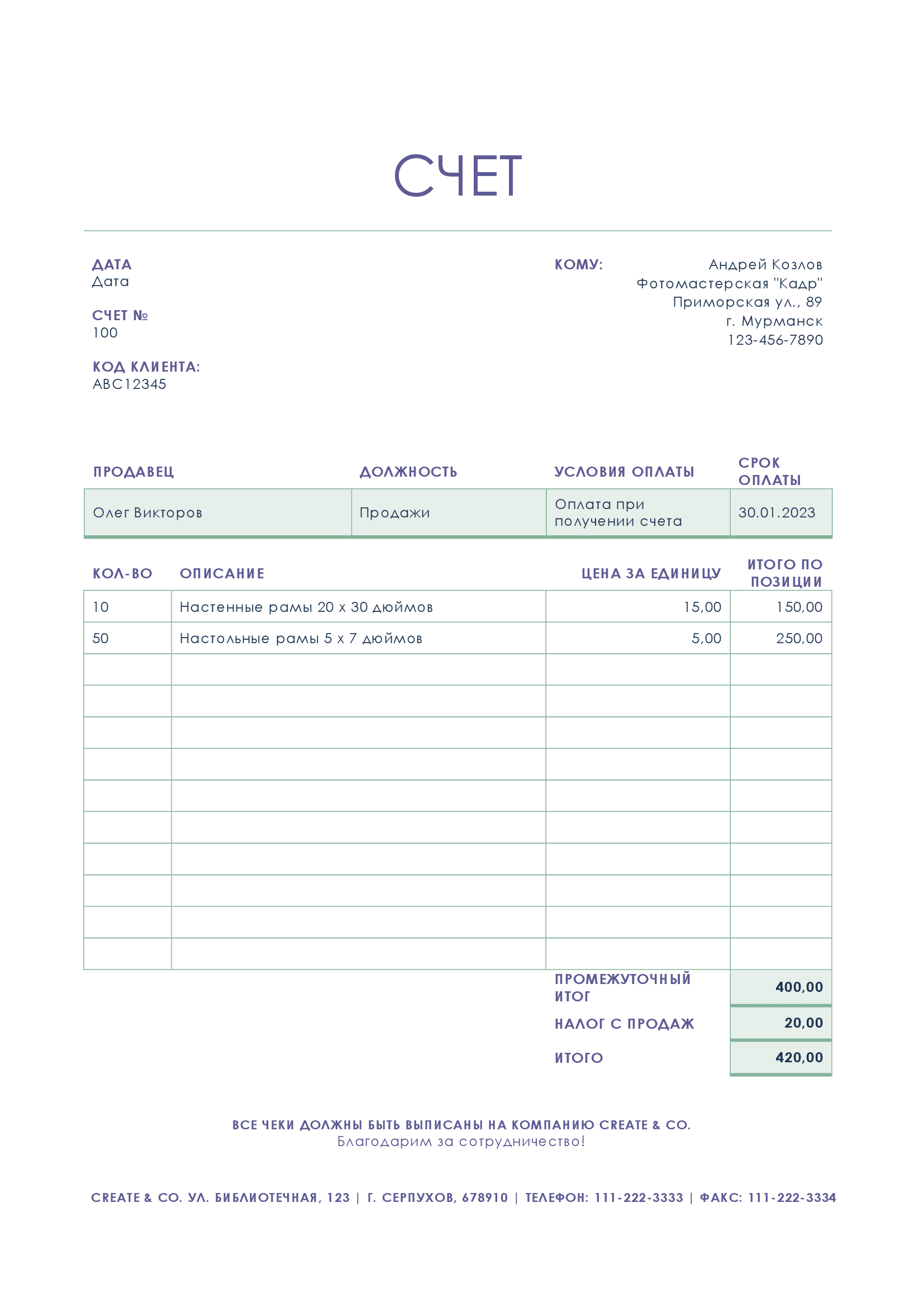


Рисунок 1.1 – Бланк со счётом

1. Перенести действие **Sequence** на рабочую панель.
2. Добавить действие **Send Hotkey**, назвать его **'Открыть окно нажатием клавиш win + r'**. Выбрать **checkbox** (флажок) **Win** и ввести **r** в поле **Key** (**Ключ**).

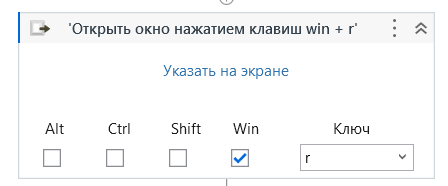


Рисунок 1.2 –Действие Send HotKey

1. Добавляем действие Type Into. В поле Text указываем в кавычках путь к файлу изображения.

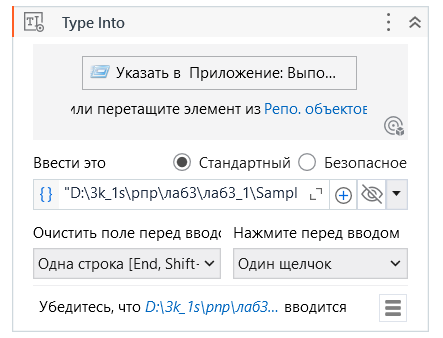


Рисунок 1.3 – Действие Type Into

1. Создаём переменную InvoiceTotal типа String и вводим ее в поле Text. После добавляем действие Get OCR Text. Кликаем по UiPath Screen OCR и нажимаем клавишу Delete, на его место помещаем действие Tesseract OCR. Открываем картинку SampleTest.png. Нажимаем на ссылку Указать на экране в действии, выделяем область на изображении. Аналогичными действиями считываем данные о покупателе.

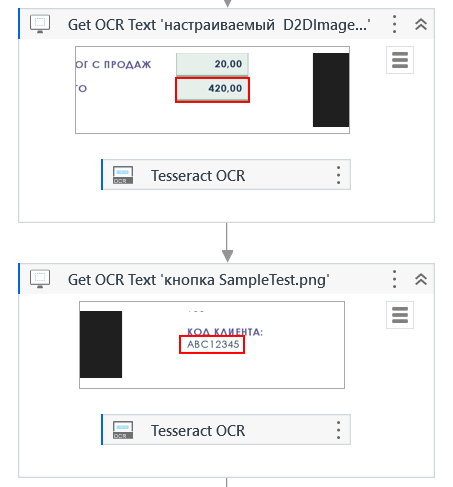


Рисунок 1.4 – Действие Get OCR Text

1. Получив все необходимые данные из счёта, формируем из них таблицу и сохраняем её в переменную ResultDT. Используя действие Write CSV записываем нашу таблицу в файл.

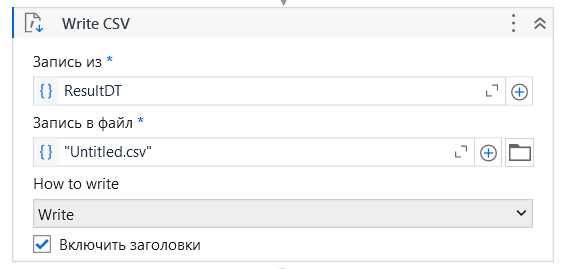


Рисунок 1.5 – Действие Write CSV

1. После выполнения процесса мы автоматически получим csv файл следующего вида.

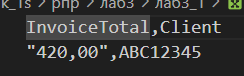


Рисунок 1.6 – Вывод в csv файле

**Задание 3-2**. Создать процесс автоматизации, который будет извлекать информацию о сотрудниках с веб-сайта и сохранять ее в файле Excel.

1. Создать процесс и назвать его **FromWebToExcel**.
2. Перенести действие **Sequence** на рабочую панель.

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

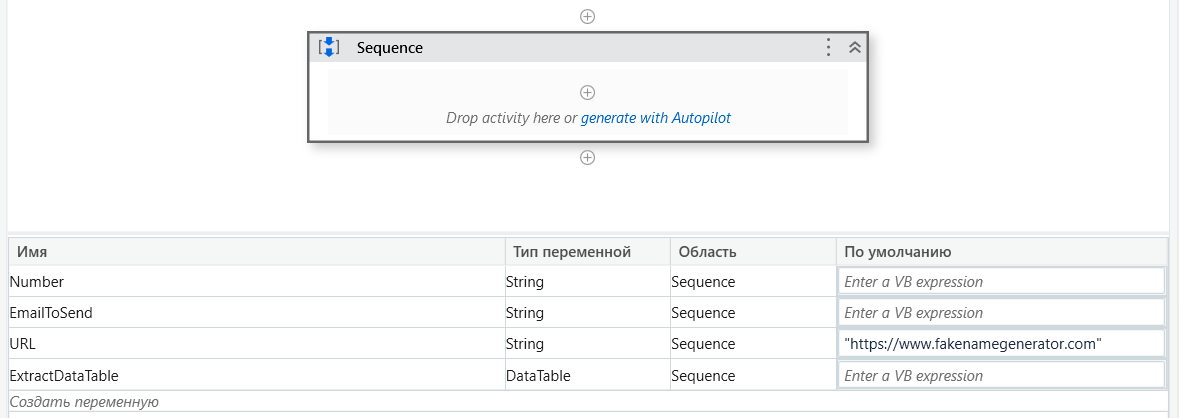


Рисунок 2.1 – Действие Sequence и созданные переменные

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**.

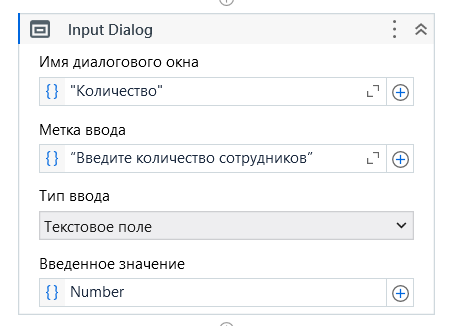


Рисунок 2.2 – Input Dialog

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** типа **String** в таблице.

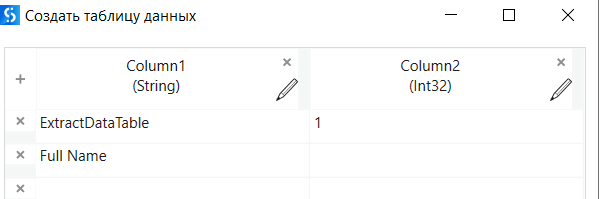


Рисунок 2.3 – Добавление таблицы

5. Добавить действие **Open Browser**.

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

6. Переименовать блок Do в **Do –** ‘**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)**.

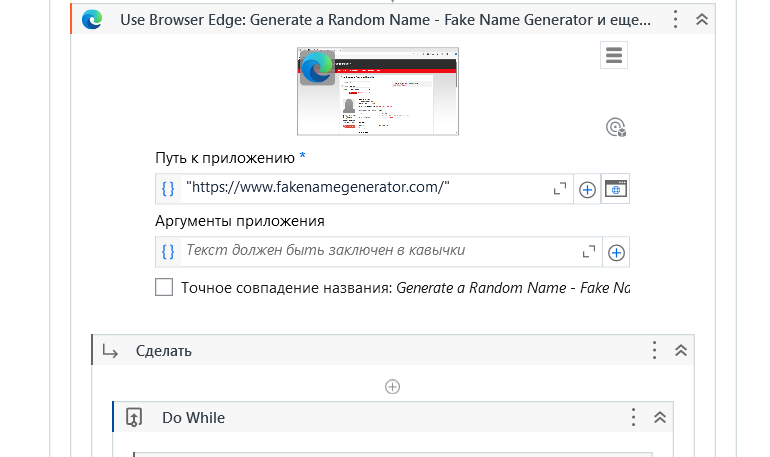
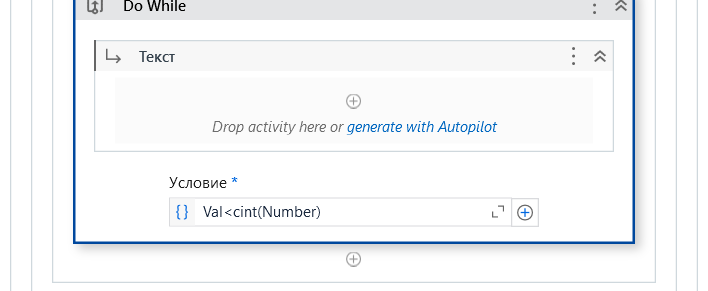
 ****

Рисунок 2.4 – Данные действия Use Brosser

Создать переменную **FullName** типа **String**.

8. Добавить действие **Assign** в последовательность.

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

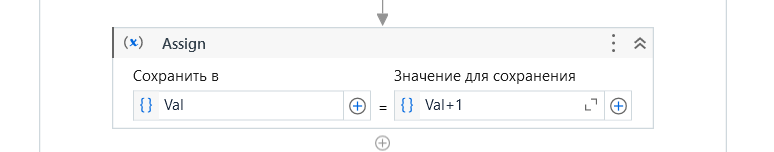


Рисунок 2.5 – Действие Assign

9. Добавить действие **Get Full Text**.

Открыть сайт **https://www.fakenamegenerator.com** в браузере Internet Explorer.

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

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

10. Добавить действие **Add Data Row.**

В свойстве **ArrayRow** 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**.

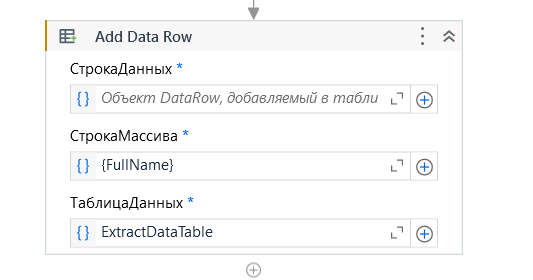


Рисунок 2.6 – Действие Add Data Row

11. Добавить действие **Click**.

Нажать ссылку **Указать элемент в браузере** и указать кнопку «Generate» на странице в браузере.

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

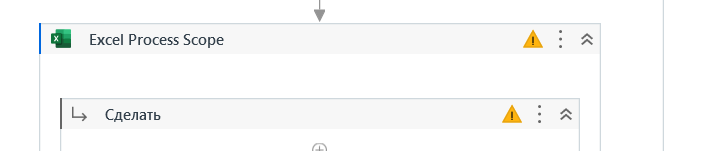


Рисунок 2.7 – Действие **Excel Application Scope**

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

13. Добавить действие **Write Range**.

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

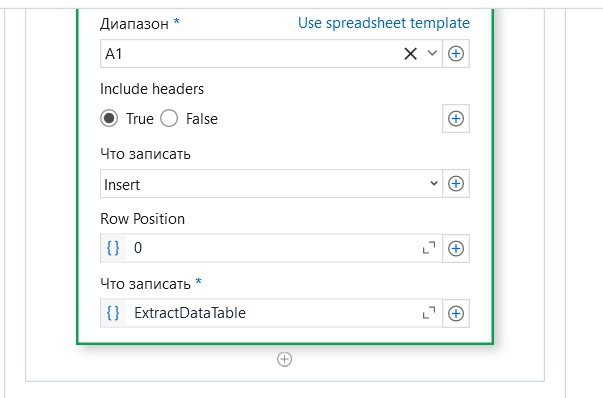
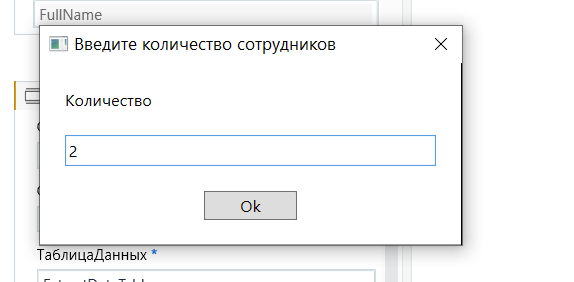
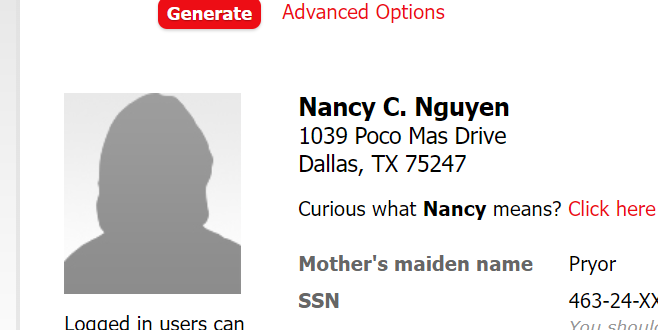
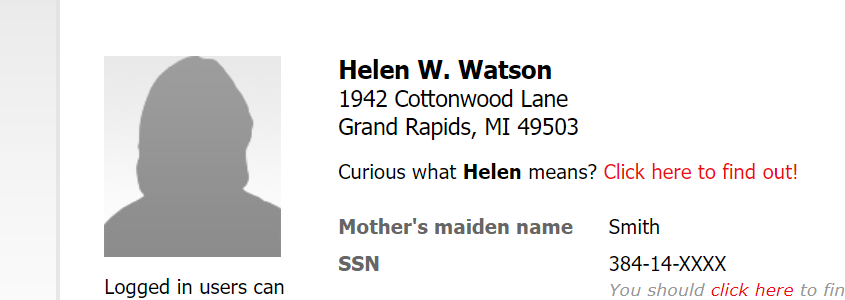


Рисунок 2.8 – Добавление условий выполнения

14. Запустить процесс автоматизации на выполнение.





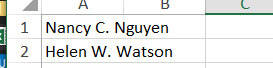


Рисунок 2.9 – Результат

**Задание 3-3**. Создать процесс автоматизации для заполнения веб-формы данными, извлеченными из таблицы Excel.

Создаём excel файл с данными, которые необходимо ввести на сайте.

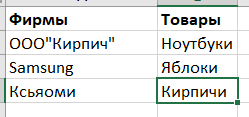


Рисунок 3.1 – Входные данные в файле Excel

Создаём страницу с двумя полями для заполнения формой.

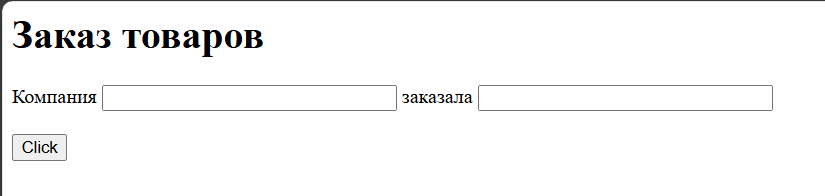


Рисунок 3.2 – Страница с формой

Для создания страницы используем язык разметки html.

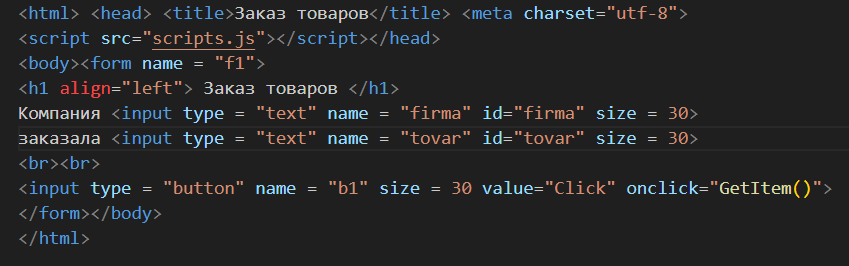


Рисунок 3.3 – Код страницы на html

Для удобного взаимодействия с формой и отслеживания введённых данных, создаём скрипт на JavaScript, который будет считывать данный полей, выводить их в консоль, а после очищать.

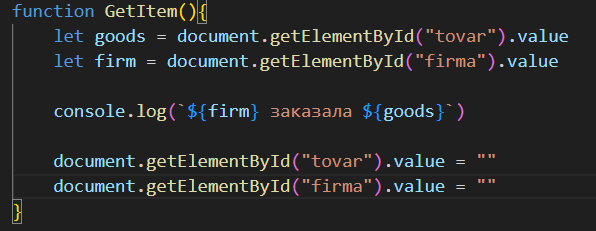


Рисунок 3.4 – Код скрипта на JavaScript

После создания сайта, запускаем его на Live Server и копируем адрес. В sequence добавляем действие open browser с ссылкой на сайт.

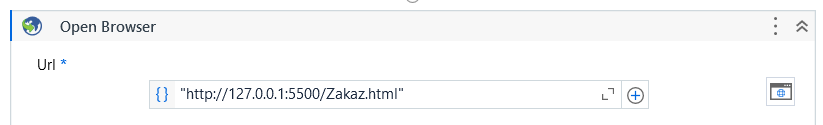


Рисунок 3.5 – Действие Open Browser

Добавляем действие Maximize Windows, для отображения страницы на весь экран.



Рисунок 3.6 – Действие Maximize Window

Считываем данные из excel и добавляем действие for each row in datatable. Внутрь добавляем действия CV Type Into. Указываем на нашем сайте поля для ввода и указываем текст для ввода. Для Этого используем переменные Company и Goods типа string, которые хранят данные из текущей строки таблицы.

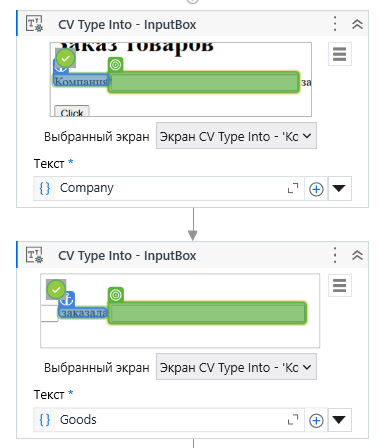


Рисунок 3.7 – Действия CV Type Into

Используем действие CV Click и указываем кнопку на сайте. Благодаря скрипту поля очистятся и мы сможем ввести следующие значения.

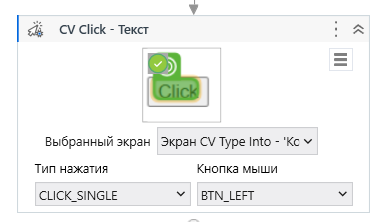


Рисунок 3.8 – Действия CV Click

После выполнения скрипта, можно зайти в браузер и посмотреть.

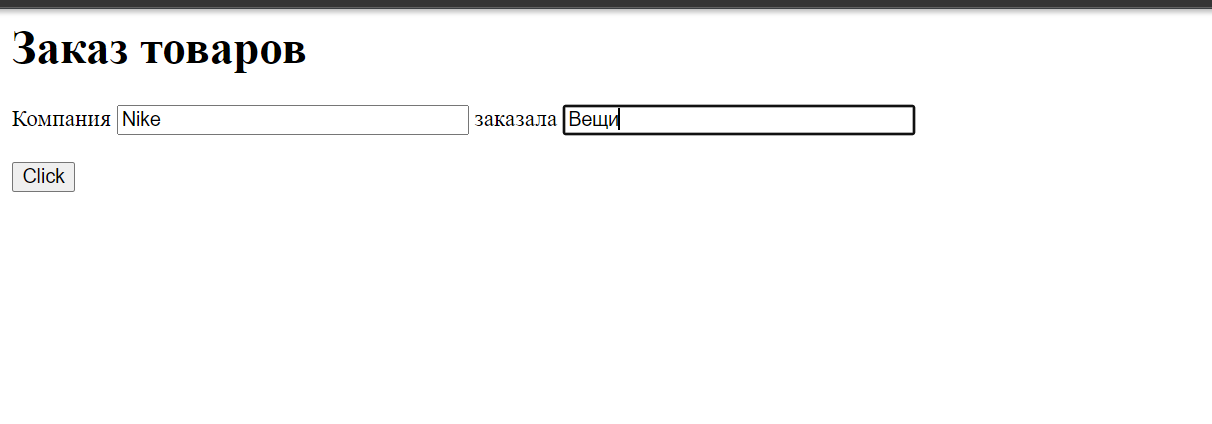


Рисунок 3.9 – Вывод в Браузере

Задание 3-4. Создать процесс автоматизации, который извлекает список статей по экономике с сайта Википедии с помощью инструмента Считывание данных (Data Scraping) и записывает список в таблицу Excel.

Открываем Chrome и переходим на сайт <https://ru.wikipedia.org>. В поле найти в Википедии вводим "экономика" и нажимаем пункт "Поиск страниц, содержащих экономика" в возникшем выпадающем списке.

Переносим последовательность Sequence на рабочую панель и добавляем в нее действие Use Browser. В поле Url помещаем URL-адрес веб-страницы с результатами поиска.

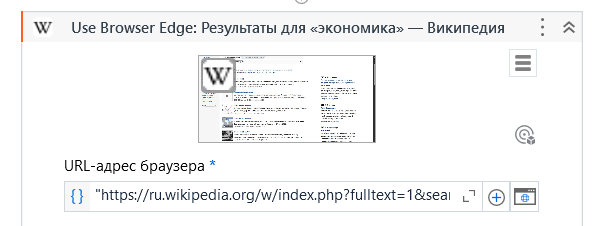


Рисунок 4.1 – Действие Use Browser

На ленте нажимаем Считывание данных. Откроется Мастер извлечения. Переходим на веб-страницу, нажимаем в мастере кнопку Далее, выбираем первый элемент, опять нажимаем Далее выбираем последний элемент на веб-странице.

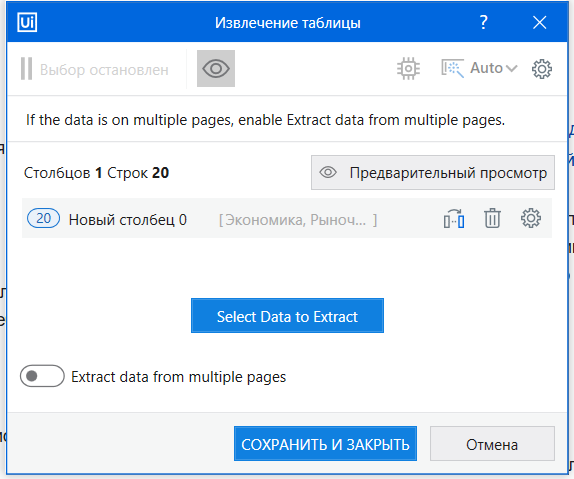


Рисунок 4.2 – Инструменты для извлечения таблицы

Нажимаем Готово. В появившемся окне Данные охватывают несколько страниц? выбираем вариант Нет.

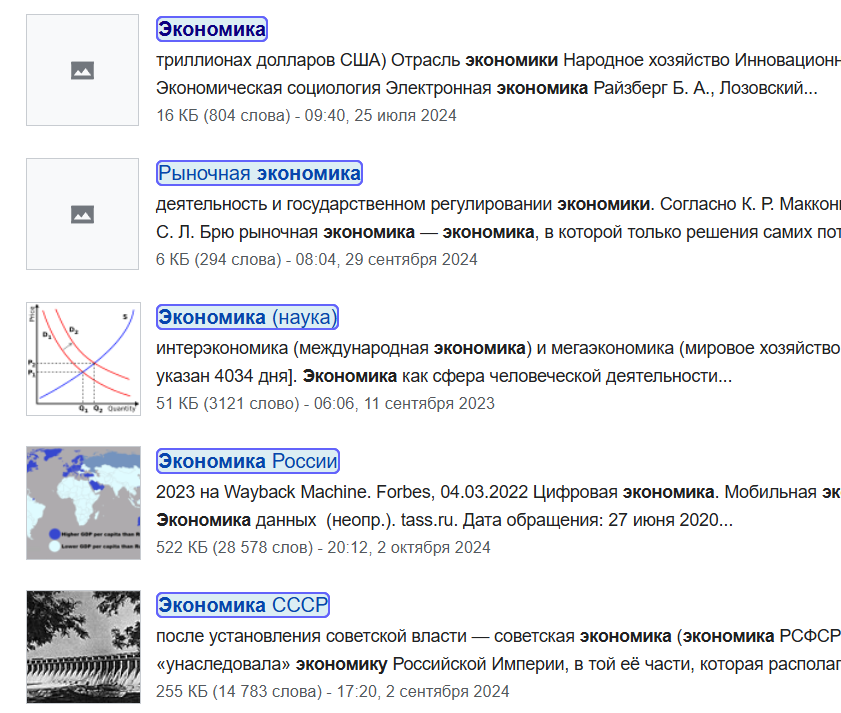


Рисунок 4.3 – Страницы сайта с полями для извлечения

Автоматически создается переменная ExtractDataTable. На панели Переменные изменяем область автоматически созданной переменной ExtractDataTable на Sequence.

Добавляем действие Excel Application Scope после раздела Считывание данных.

На панели свойств в поле Путь к рабочему журналу вводим имя файла "web\_scraping.xlsx". После выполнения проекта в папке проекта будет создан файл с этим именем.

В контейнере Сделать (Do) добавляем действие Write Range и на панели свойств в поле DataTable записываем переменную ExtractDataTable. Ставим флажок Добавить заголовки.

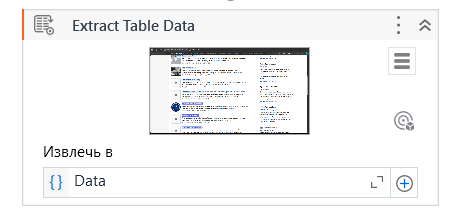


Рисунок 4.4 – Действие Extract Table Data

После выполнения процесса наш Excel файл сам заполниться данными со страницы.

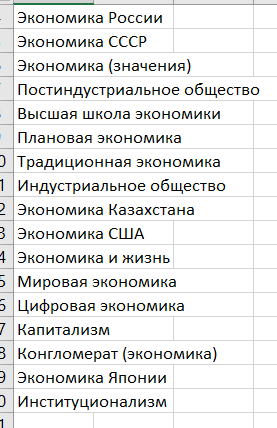


Рисунок 4.5 – Вывод в файле Excel

**Задание 3-5**. Создать процесс автоматизации, который получает из диспетчера учетных записей Windows данные пользователя для подключения к почтовому ящику, а затем отправляет письмо на этот же адрес с помощью действия **Send SMTP Mail Message**. Затем процесс автоматизации должен получить последние полученные 5 писем из этого ящика с помощью действия **Get IMAP Mail Messages**, сохранить текст в файле и вложения последнего пришедшего письма.

Открываем приложение Диспетчер учетных данных используя Пуск / Панель управления / Диспетчер учетных данных. Нажимаем на ссылку Добавить общие учетные данные. В поле Адрес в Интернете или сети вводим название учетных данных. В полях Имя пользователя и Пароль вводим адрес электронной почты и пароль.

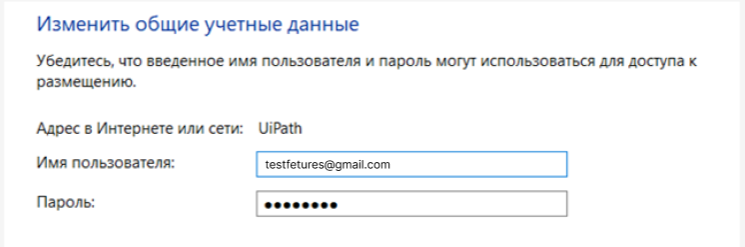


Рисунок 5.1 – Окно изменения учетных данных

Перенеосим последовательность Sequence на рабочую панель. Создать переменную mailUser типа String, переменную mailPass типа SecureString. На панели Проект нажимаем кнопку Управление пакетами. Переходим в раздел Все пакеты, выбираем пакет UiPath.Credentials.Activities, нажимаем кнопку Установить, затем кнопку Сохранить.

Добавляем действие Get Secure Credential в последовательность. В свойстве Target указываем имя учетной записи из диспетчера учетных данных, в Password указываем переменную mailPass, а в поле Username указать переменную mailUser.

Добавляем действие Send SMTP Mail Message. В свойствах Port и Server, "smtp.gmail.com", 465 для Gmail почты.

В свойстве Email указываем переменную mailUser, в свойство Password записываем пароль, полученный из диспетчера учетных данных, new System.Net.NetworkCredential(mailUser, mailPass).Password.

В поле To вводим переменную mailUser. В поле Subject указываем тему письма "Test Message", в поле Body вводим текст сообщения "Hello from Mail Robot!".

Нажимаем в действии на ссылку Прикрепить файлы (Attach Files). В появившемся окне в поле Value указываем название файла с картинкой "imageUiPath.jpg".

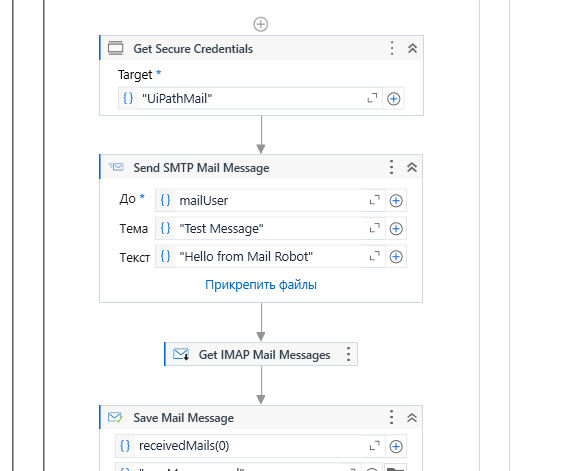


Рисунок 5.2 – Действие Send SMTP Mail Message

Создаём переменную receivedMails типа List<MailMessage>

Далее добавляем действие Get IMAP Mail Messages. В свойствах Port и Server указываем IMAP сервер и IMAP порт для нужного почтового сервера "imap.gmail.com", 993 для Gmail почты. В свойстве Email указываем переменную mailUser, в свойство Password записываем пароль, полученный из диспетчера учетных данных new System.Net.NetworkCredential(mailUser, mailPass).Password. Ставим отметку в поле OnlyUnreadMessages. В свойстве Top пишем 5. В свойстве Messages указываем переменную receivedMails.



Рисунок 5.3 – Действие Get IMAP Mail Messages

Добавляем действие Save Mail Message. В свойстве FilePath указываем файл, в который будет сохранено сообщение ("newMessage.eml"). В свойстве MailMessage указываем receivedMails(0).

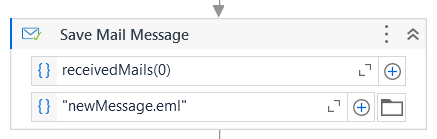


Рисунок 5.4 – Действие Save Mail Message

Добавляем действие Save Attachments. В свойстве FolderPath указываем путь к папке, в которой будут сохранены вложения ("attachments"). В свойстве Message указываем сообщение (receivedMails(0)).

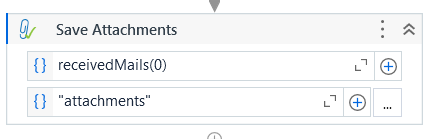


Рисунок 5.5 – Действие Save Attachments

Для успешного и правильного подключения к электронной почте нужно разрешить для нее доступ и использовать пароли приложений для безопасности. Требуется зайти в аккаунт почты. Выбрать раздел «Пароли приложений» в разделе настройки двухфакторной защиты. После указания названия приложения, платформа генерирует пароль для доступа, к электронной почте который необходимо использовать для аутентификации.

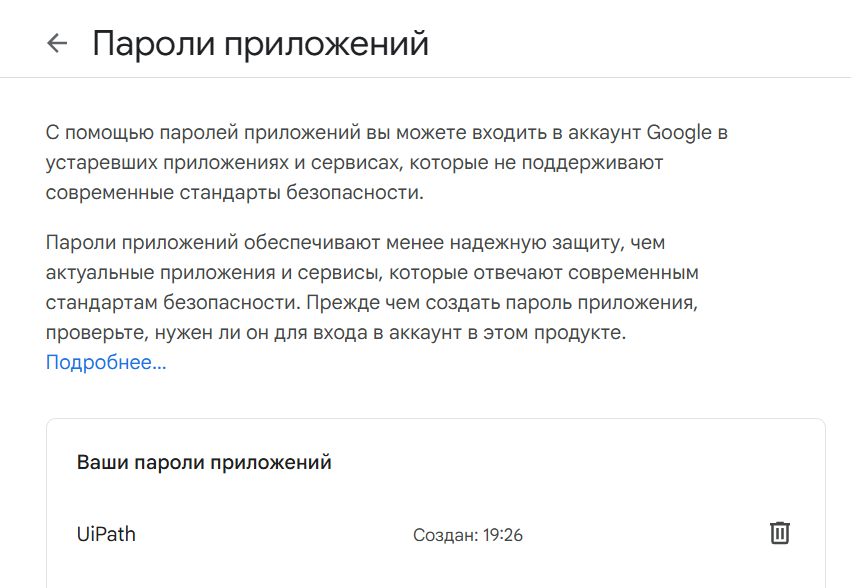


Рисунок 5.6 – Настройка пароля приложения

После выполнения процесс, на указанный адрес должно прийти сообщения с наши содержанием.

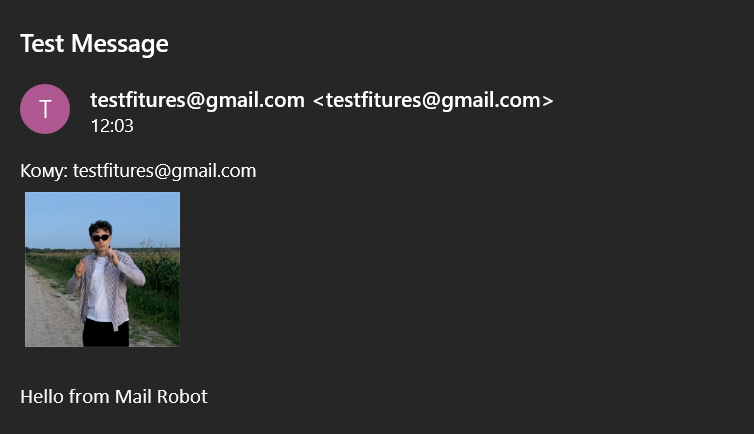


Рисунок 5.7 – Сообщение