Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Разработка программных роботов

Студент: Немкович А.В.

ФИТ 3 курс 1 группа

Преподаватель: Сазонова Д.В.

Минск 2024

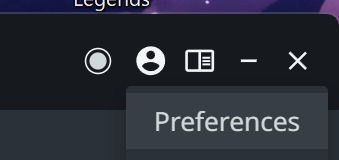
Лабораторная работа № 6. Использование шаблона Robotic

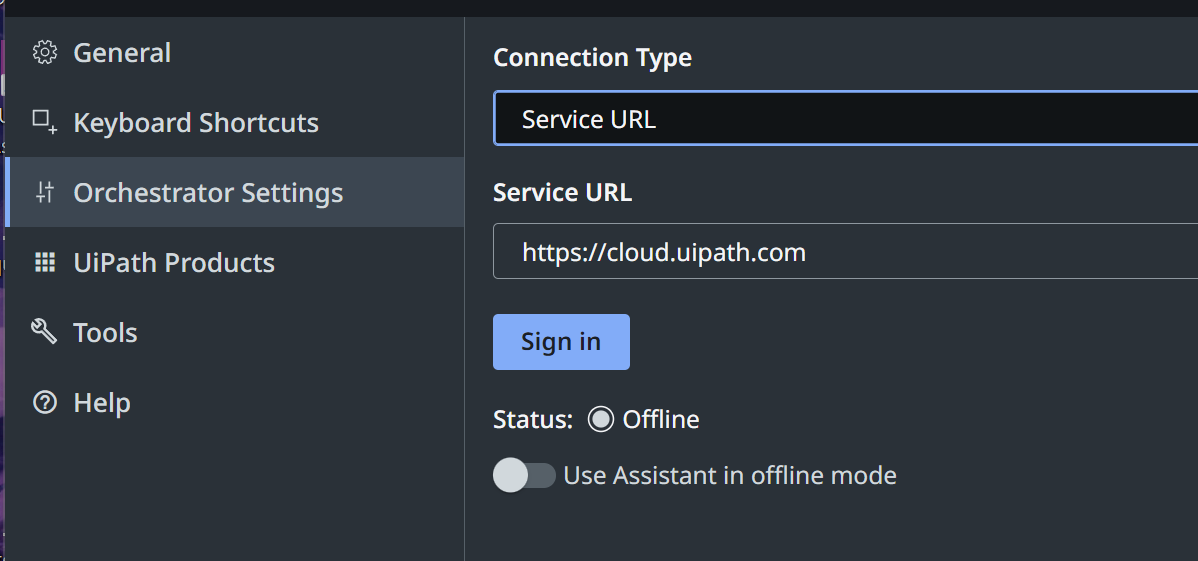
Enterprise Framework

**ReFramework** – это **шаблон**, который используется для создания рабочих процессов автоматизации.

**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.

Предварительно следует убедиться, что все процессы автоматизации отключены от Оркестратора. Надо запустить UiPath Assistant и в настройках Оркестратора нажать кнопку Отключить для процессов автоматизации, подключенных к Оркестратору.





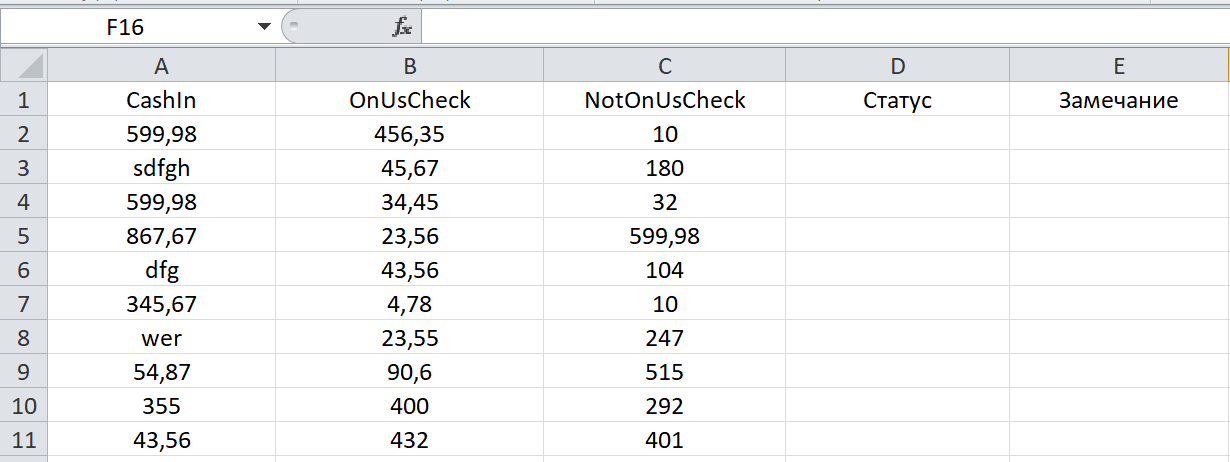
Тут было sign out или что-то в этом роде

Для работы необходимо скачать шаблон более старой версии, поскольку в новых версиях отсутствуют необходимые компоненты. Нужная версия доступна на сайте github: <https://github.com/UiPath/ReFrameWork>

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 данном описании предполагается использование классического дизайна UiPath Studio.

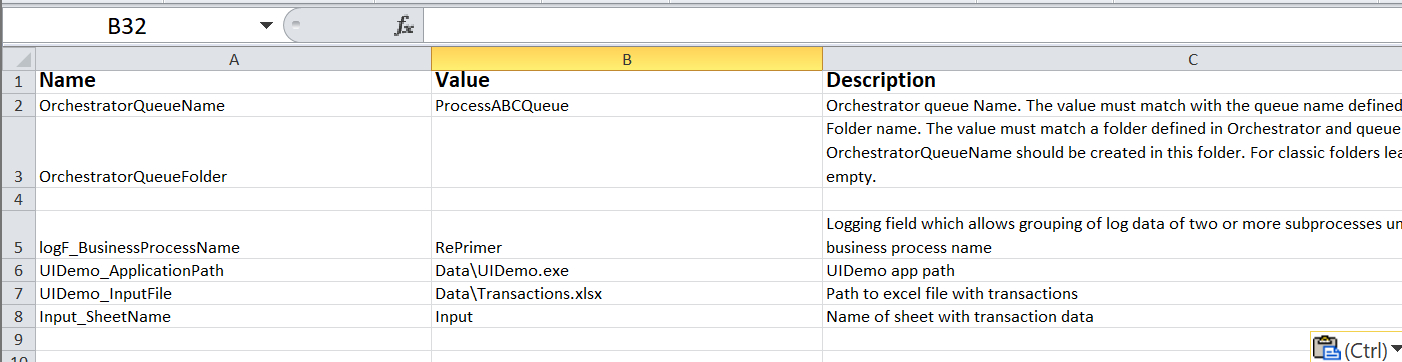
1. В UiPath Studio создать новый процесс, используя шаблон Robotic Enterprise Framework (Старт/Роботизированная инфраструктура предприятия), и назвать его ReFrameworkPrimer.

2. 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.

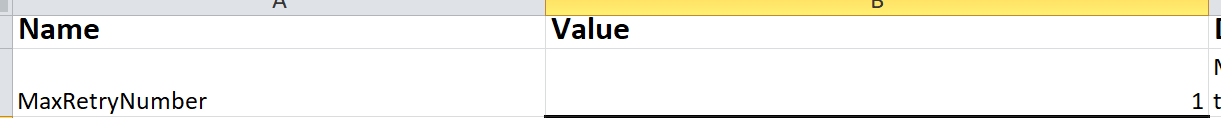
3. Открыть конфигурационный файл Data/Config.xlsx.

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 |



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



4. Открыть рабочий процесс Main.xaml и раскрыть состояние Init (Инициализация) дважды щелкнув по нему.

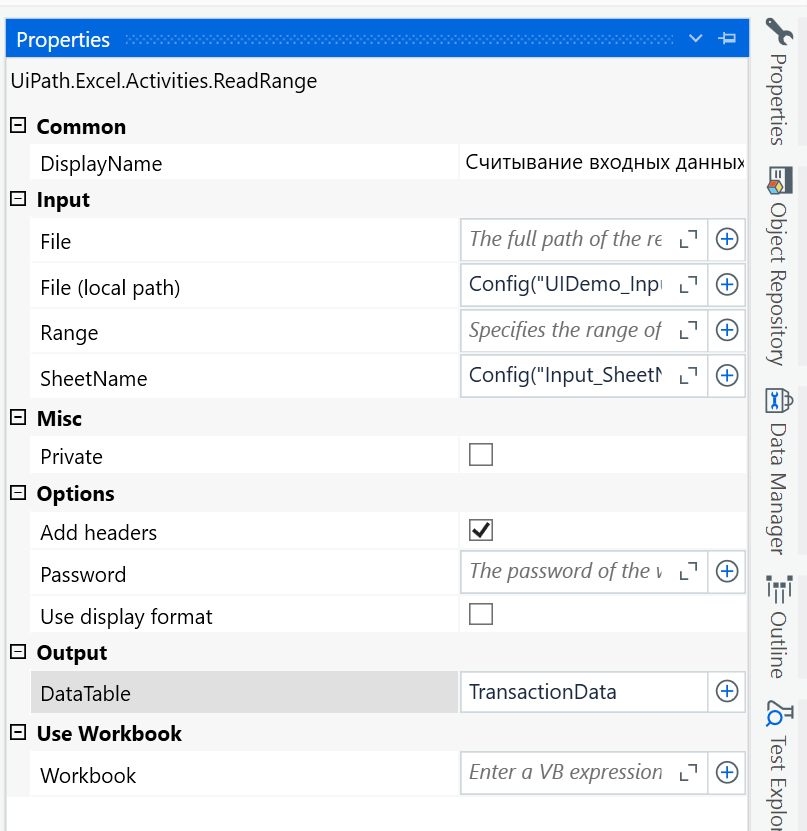
Раскрыть процесс if first run – read Config file двойным щелчком и, так как очередь использоваться не будет, удалить последовательность c названием Save job arguments Config. – я не нашла такого

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

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

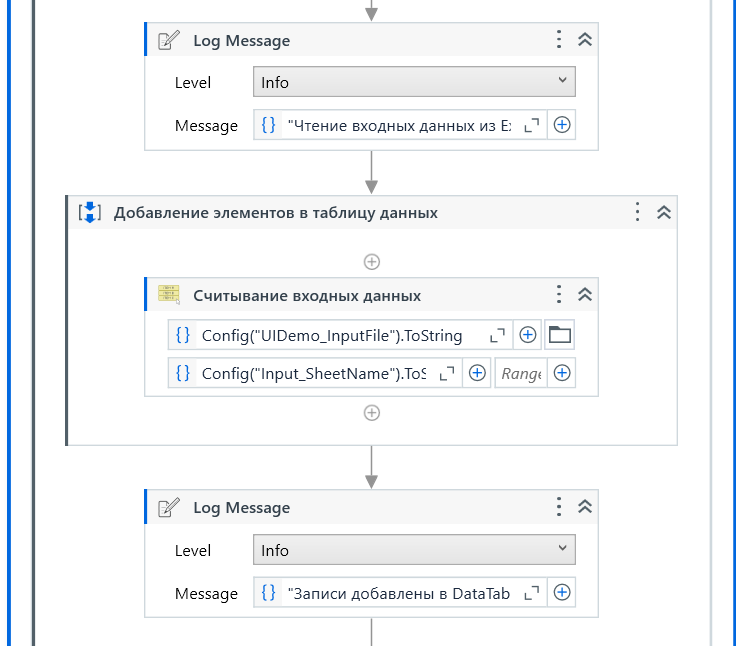
Установить следующие свойства:

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



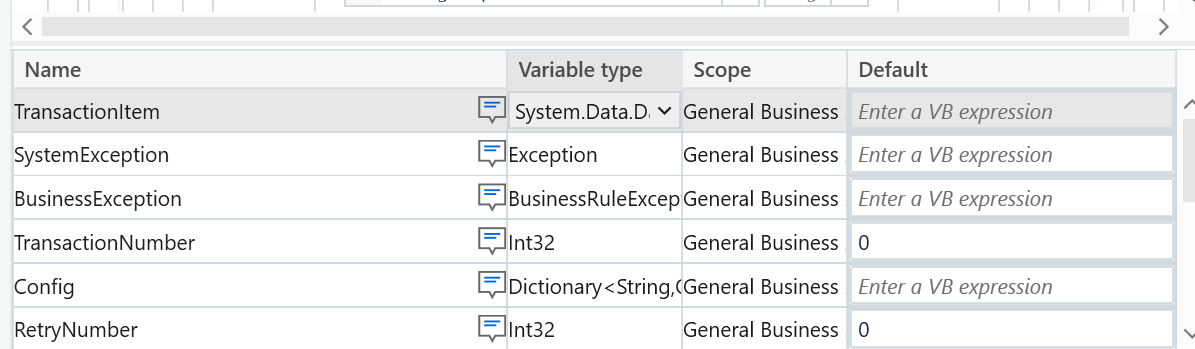
7. Добавить действия Log Message до и после действия Read Range.

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



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

(для выбора типа в поле Тип переменной => Выбор типов => System.Data.DataRow).



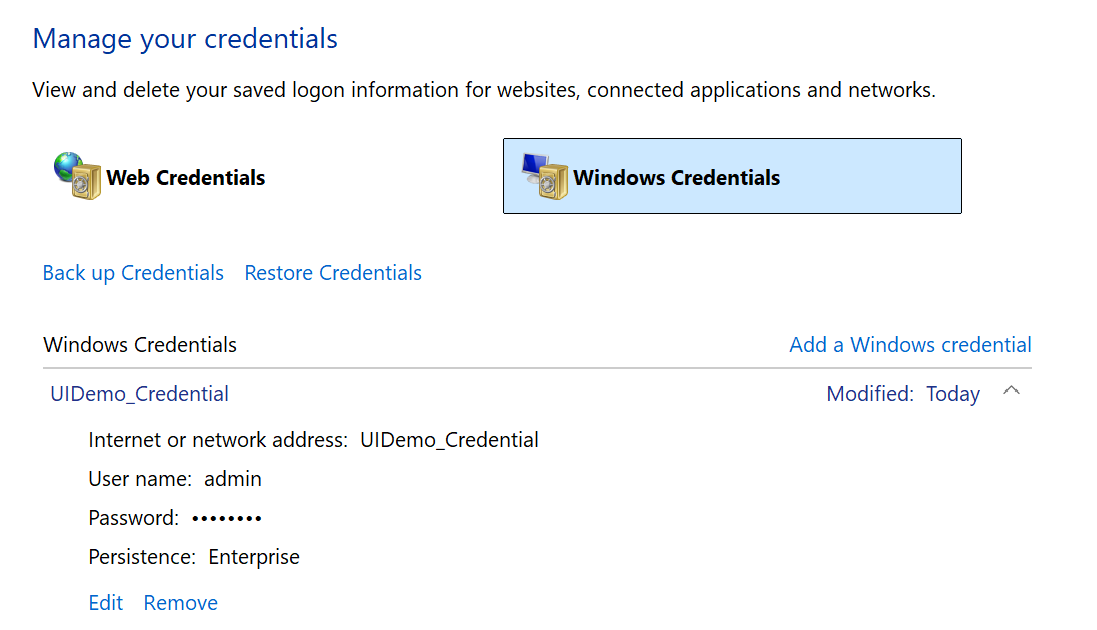
9. Открыть приложение Диспетчер учетных данных

(Пуск / Панель управления / Диспетчер учетных данных).

Выбрать Учетные данные Windows. Нажать на ссылку Добавить общие учетные данные.

В поле Адрес в Интернете или сети ввести название учетных данных, по которым позже будет происходить их получение, например, UIDemo\_Credential.

В поле Имя пользователя ввести admin, в поле Пароль ввести password.



10. Открыть на вкладке Проект в нижней части экрана рабочий процесс Framework/InitAllApplications.xaml.

Перенести в последовательность рабочий процесс Framework/GetAppCredentials.xaml и разместить его после блока Log App Initialization message.

В появившемся действии Invoke... нажать кнопку Импортировать аргумент (Import Arguments) и изменить значение аргумента in\_Сredential на "UIDemo\_Credential".

В поле Value аргумента out\_Username нажать сочетание клавиш Ctrl+K, ввести имя переменной UserName и нажать Enter.

Проделать то же самое для поля out\_Password (ввести имя переменной Password).

*Комбинация клавиш Ctrl+K позволяет автоматически создать переменные с требуемыми типами данных.*

11. В рабочем процессе GetAppCredentials.xaml открыть панель аргументов и удалить в аргументе in\_Credential значение по умолчанию.

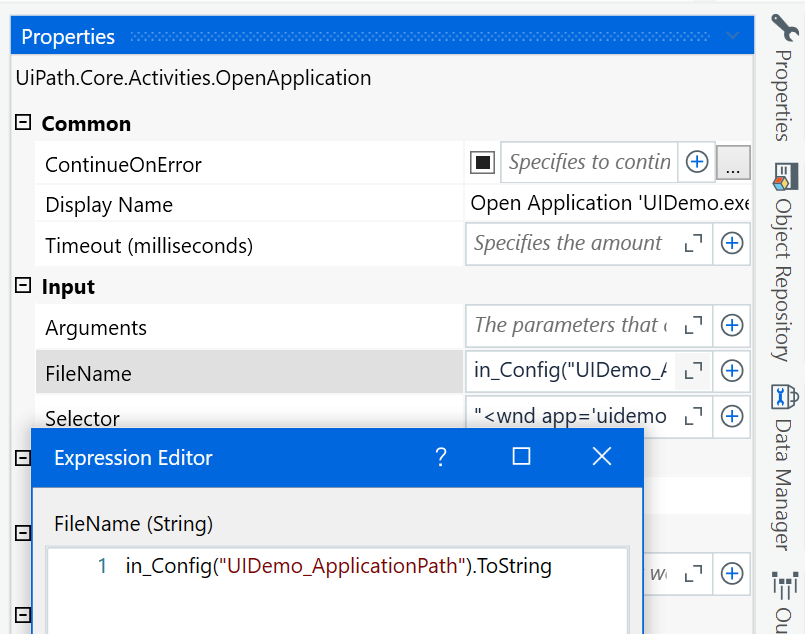
12. Вернуться в рабочий процесс InitAllApplications.xaml.

Добавить действие Open Application.

Запустить UIDemo.exe из папки Data.

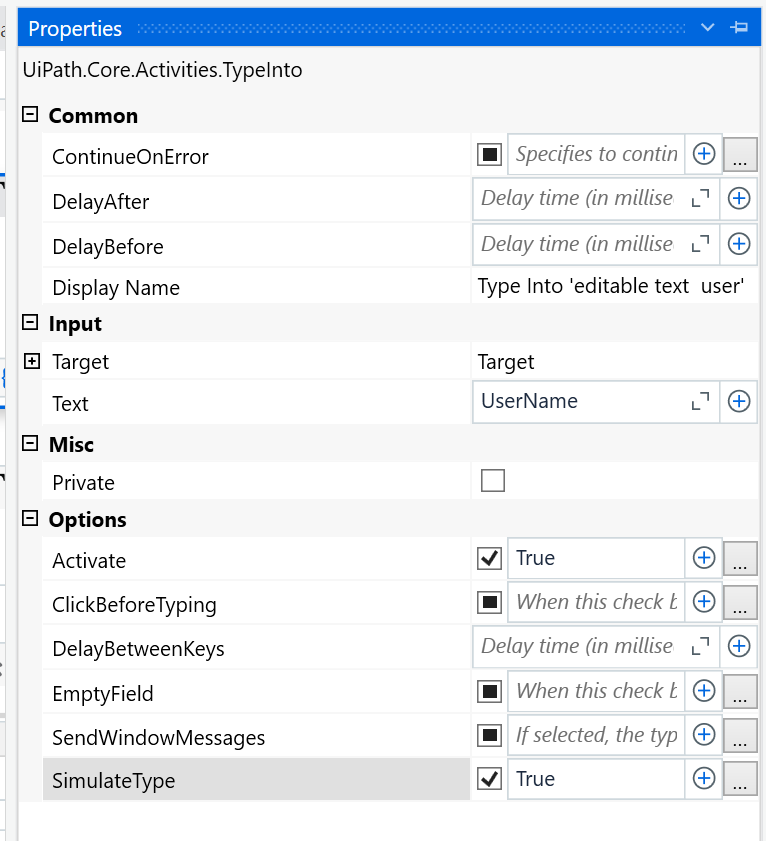
Нажать на ссылку Указать на экране и отметить окно UIDemo.exe.

Ввести текст in\_Config("UIDemo\_ApplicationPath").ToString в поле FileName.



13. Добавить действия Type Into, Type Secure Text, Click в последовательность Do.

14. В действии Type Into нажать на ссылку Указать на экране, щелкнуть по первому полю в приложении UIDemo.exe, в поле Text указать переменную UserName, поставить галочку в свойстве Simulate Type.



15. В действии Type Secure Text нажать на ссылку Указать на экране, щелкнуть по второму полю приложения UIDemo.exe, добавить в поле Secure Text переменную Password, поставить галочку в свойстве Simulate Type.

16. В действии Сlick нажать на ссылку Указать на экране, щелкнуть по кнопке Log in в приложении UIDemo.exe, поставить галочку в свойстве Simulate Click.

17. Добавить в последовательность действие Element Exists.

В приложении UIDemo.exe ввести в первое поле *admin*, во второе –*password* и нажать Log In. Появится окно UiDemo.

В действии нажать на Указать на экране, щелкнуть по окну приложения.

В свойстве Exists с помощью клавиш Ctrl+K ввести имя переменной UIDemo\_LoginSuccess.

18. Добавить действие If.

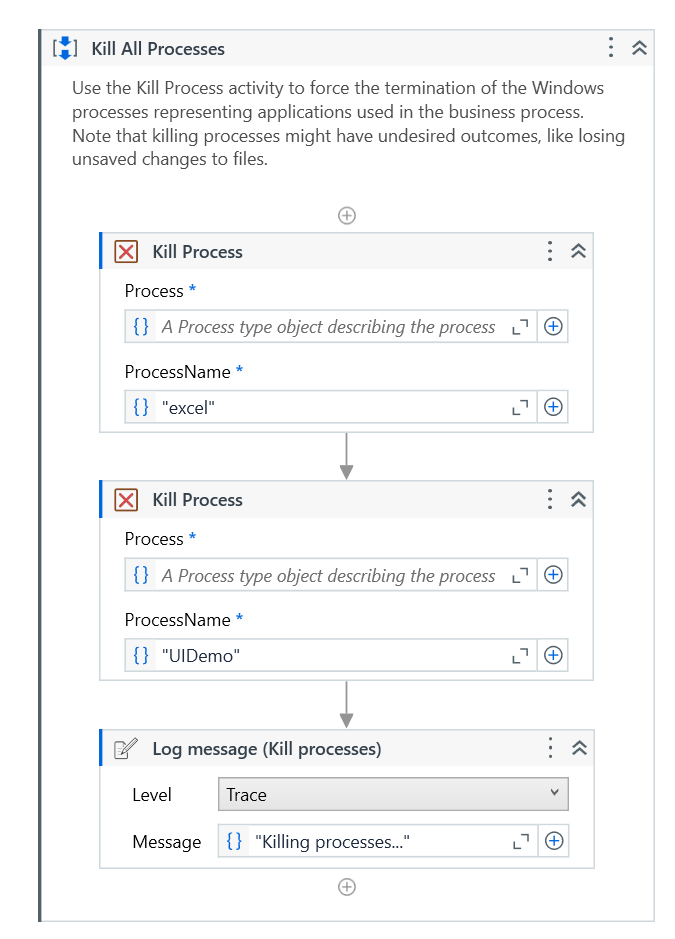
В поле Condition ввести переменную UIDemo\_LoginSuccess.

Перенести в блок Then действие Log Message и добавить в поле Message текст "UIDemo\_приложение успешно открыто".

19. Открыть рабочий процесс Framework/KillAllProcesses.xaml.

Перенести два действия Kill Process.

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



20. Открыть рабочий процесс Framework/GetTransactionData.xaml.

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

21. Удалить из рабочего процесса действие Get Transaction Item.

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

23. В блоке Then изменить название действия Assign TransactionID на Assign TransactionItem.

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

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

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

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

(действие Retrieve Data => действие Should Stop or Get Next => блок Else => блок GetTransactionData, двойной щелчок по блоку Try).

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

27. В блоке Then действия Should Stop or Get Next в действии Assign с названием End Process перезаписать в правом поле значение Nothing на Nothing.

28. Справа в блоке Else развернуть секцию Exception, найти действие Assign с названием End Process и перезаписать в правом поле значение Nothing на Nothing.

29. Вернуться в самое начало рабочего процесса Main.xaml и дважды щелкнуть по переходу New Transaction.

Изменить условие Condition на TransactionNumber < TransactionData.Rows.Count.

30. На вкладке Проект в нижней части экрана найти рабочий процесс Process.xaml и открыть его.

На панели Аргументы изменить тип аргумента in\_TransactionItem на DataRow.

31. Перенести в рабочий процесс действие Log Message и добавить в поле Message текст "Начало процесса".

32. 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")).

33. Перенести в блок Then действие Attach Window, нажать на ссылку Указать на экране и щелкнуть по окну приложения UIDemo.

34. 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.

35. 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.

36. 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.

37. Добавить действие Click,

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

38. В блок Else перенести действие Throw,

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

39. Открыть процесс Framework/SetTransactionStatus.xaml.

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

40. 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.

41. В блок Then перенести последовательность Sequence и поместить в нее действие Write Cell из раздела Рабочий журнал, указать следующие свойства:

|  |  |
| --- | --- |
| Свойство | Значение |
| WorkbookPath | in\_Config("UIDemo\_InputFile").ToString |
| SheetName | in\_Config("Input\_SheetName").ToString |
| Cell | "D"+ (io\_TransactionNumber+2).ToString |
| Text | "Успешно" |

42. Скопировать действие Write Cell и поместить за первым.

Изменить в свойстве Cell имя ячейки на "E"+ (io\_TransactionNum-ber+2).ToString и сообщение в свойстве Text на "Обработано!".

43. Вернуться в начало рабочего процесса Framework/SetTransactionSta-tus.xaml.

Дважды щелкнуть по последовательности Handle Business Rule Exception.

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

В блоке Then удалить действие Set Transaction Status.

44. В блок Then перенести последовательность Sequence и поместить в нее действие Write Cell из раздела Рабочий журнал, указать следующие свойства:

|  |  |
| --- | --- |
| Свойство | Значение |
| WorkbookPath | in\_Config("UIDemo\_InputFile").ToString |
| SheetName | in\_Config("Input\_SheetName").ToString |
| Cell | "D"+ (io\_TransactionNumber+2).ToString |
| Text | "Пропущено" |

45. Скопировать действие Write Cell и поместить за первым.

Изменить в свойстве Cell имя ячейки на "E"+ (io\_TransactionNumber+2).ToString и сообщение в свойстве Text на in\_BusinessRuleException.Message.

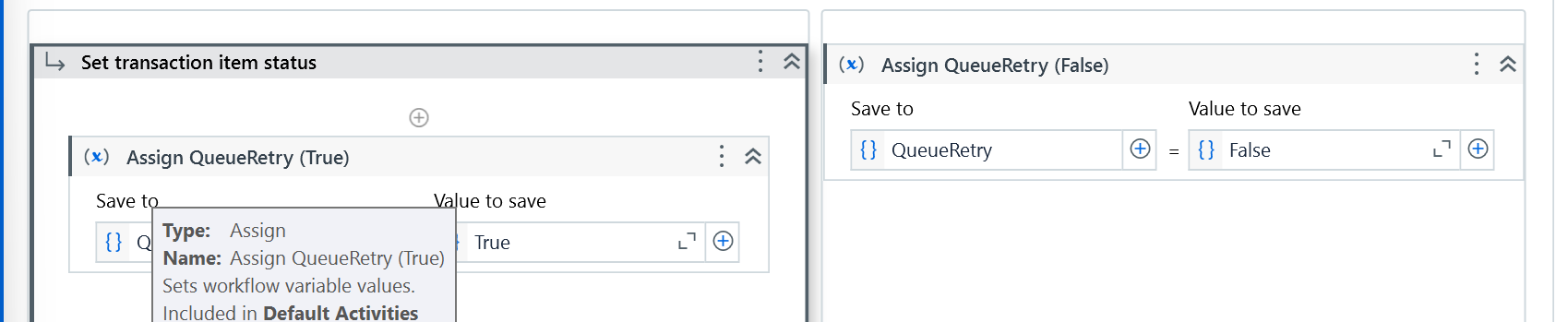
46. Вернуться в начало рабочего процесса Framework/SetTransactionStatus.xaml.

Перейти в последовательность Handle System Error.

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

47. В блоке Then удалить действие Set Transaction Status и действие Assign RetryNumber from Queue.

Удалить действия Assign с названиями Set queueRetry из блоков Then и Else.



На панели Переменные установить значение переменной queueRetry по умолчанию False.

48. Перенести в блок Then действие Write Cell из раздела Workbook.

Указать следующие свойства:

|  |  |
| --- | --- |
| Свойство | Значение |
| WorkbookPath | in\_Config("UIDemo\_InputFile").ToString |
| SheetName | in\_Config("Input\_SheetName").ToString |
| Cell | "D"+ (io\_TransactionNumber+2).ToString |
| Text | "Неудача" |

49. Скопировать действие Write Cell и поместить за первым.

Изменить в свойстве Cell имя ячейки на "E"+ (io\_TransactionNumber+2).ToString и сообщение в свойстве Text на in\_SystemError.Message.

50. Открыть Main.xaml, развернуть состояние Process Transaction (Обработка транзакций).

В блоке Try найти вызов рабочего процесса Process.xaml, нажать кнопку Импортировать аргументы, в аргументе in\_TransactionItem указать переменную TransactionItem и изменить тип на DataRow.

51. Ниже развернуть блок Finally, найти вызов рабочего процесса Framework/ SetTransactionStatus.xaml.

Нажать кнопку Импортировать аргументы, изменить значение аргумента in\_TransactionItem c Nothing на TransactionItem.

52. Открыть рабочий процесс Framework/CloseAllApplication.xaml.

Перенести два действия Kill Process.

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

53. Вернуться к Main.xaml. Запустить процесс автоматизации на выполнение.

В появившемся окне ввести имя *admin* и пароль *password*.

Проверить содержимое файла Excel.

Проанализировать сообщения на панели Вывод.

54. Для того, чтобы увидеть обработку системного исключения, надо переключиться на другую вкладку в приложении UIDemo либо закрыть это приложение во время обработки транзакции.

