



ІТМО

Роботизация процессов автоматизации (RPA)

Выполнил: Васильев Максим, М4140

Руководитель: Пастухов Денис

09.02.2024 г.



- RPA - технология, основанная на использовании ботов для решения рутинных бизнес-задач.
- Примеры работы:
 - desktop приложения
 - web страницы
 - файловая система
- Важность:
 - доверить людям более творческую работу
 - сократить возможные человеческие ошибки
 - повысить общую эффективность работы

- Automagica
 - широкий выбор сфер автоматизации
 - коммерческий продукт
- Robocorp
 - широкий выбор сфер автоматизации
 - необходима подписка для оркестрации ботов через облачный сервис Robocorp
- Сторонние open source библиотеки
 - ограниченные области применения



1. Провести анализ и осуществить выбор технологической платформы, на базе которой можно построить модуль RPA с поддержкой API для включения в интеграционную структуру поставляемого заказчику программного комплекса.
2. Спроектировать основные компоненты подключаемого интеграционного модуля
3. Реализовать основные компоненты подключаемого интеграционного модуля



Задача 1. Выбор платформы



- Были проанализированы варианты: RPA-Python, Robot Framework, Pywinauto, Selenium, TagUI, BotCity, Automagica, Robocorp, Open RPA, UI.Vision, Taskt, Pyautogui.
- Был выбран Robocorp, потому что:
 - нет ограничений по лицензированию для дальнейшего переиспользования
 - богатый выбор сфер автоматизации
 - активное развитие проекта
 - большое сообщество разработчиков

Задача 1.5. Базовые сценарии



- Была выполнена реализация основных роботизированных сценариев, которые планировалось использовать в качестве альтернативы для классических интеграционных сценариев с помощью взаимодействия через API:
 - Макрос взаимодействия с интегрируемой системой с помощью работы через пользовательский интерфейс веб-приложения через запуск скрипта с последующим выполнением последовательности нажатия элементов пользовательского интерфейса для выполнения необходимых команд для получения требуемого результата в бизнес-сценарии.
 - Аналогичный макрос для работы с десктопным приложением Windows.

Задача 2. Проектирование



- Было выполнено проектирование системы для поддержки следующих функциональных возможностей:
 - Хранение и иные операции над списком макросов.
 - Публикация выбранного макроса на удаленной машине.
 - Запуск выбранного макроса на удаленной машине.
 - Управление расписанием выполнения макросов в привязке к размещению на удаленных машинах.
 - Панель управления макросами для упрощения администрирования, включая мониторинг и развертывание макросов.

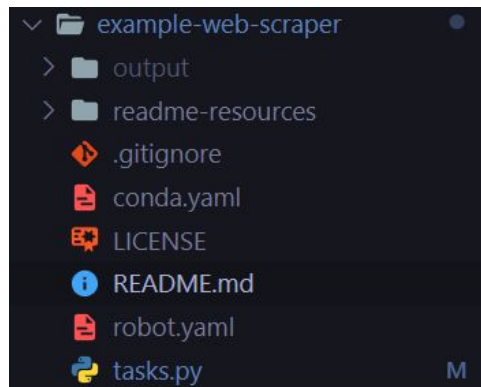
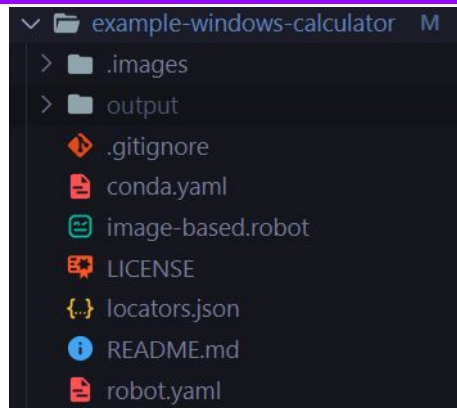
Задача 2. Проектирование



- В качестве основных модулей проектируемой системы были выделены:
 - Ядро системы, поддерживающее хранение данных по макросам, с внешним API и пользовательским интерфейсом, управление агентами на удаленных машинах.
 - Агент для публикации, запуска и администрирования макросов на удаленной машине с API используемым для интеграции с ядром системы.

Задача 3. Реализация (робот)

- Скрипт логики робота (.py или .robot)
- Конфигурация окружения (conda.yaml)
- Конфигурация робота (robot.yaml)
- Для тех роботов, которые взаимодействуют с UI - список локаторов и непосредственно локаторы (.json, .images)
- Дополнительная информация (README.md, LICENSE, .gitignore)



Задача 3. Реализация (API)

- root адрес с вводной информацией
- список всех роботов
- информация о роботе по его id
- запуск робота по его id
- удаление робота по его id





- Выполнен анализ среди большого количества альтернатив, предложены критерии выбора и осуществлен быстрый переход от первичного анализа к более детальной проработке.
- Достаточно сложная система была декомпозирована на основные компоненты.
- Была выполнена разработка первых версий компонентов для прототипирования и дальнейшей проработки и совершенствования разрабатываемой системы.

**Спасибо
за внимание!**

it'sMO *re than a*
UNIVERSITY