Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»

Институт №3 «Системы управления, информатика и электроэнергетика»

Кафедра №307 «Цифровые технологии и информационные системы»

Отчёт о выполнении работы  
по предмету «Архитектура информационных систем»

«Робот-доставщик»

Выполнили:

Студенты группы М3О-221Б-21

Прокофьев Сергей Алексеевич

Подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Сухарев Александр Игоревич

Подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Принял:

Преподаватель кафедры 307

Максимов Алексей Николаевич

Подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Оглавление**

[Постановка задачи 3](#_Toc137810713)

[Архитектура программы 4](#_Toc137810714)

[Заключение 9](#_Toc137810715)

[Источники 10](#_Toc137810716)

# Постановка задачи

Цель данной работы состоит в разработке программы для робота-доставщика напитков, которая позволит пользователям заказывать напитки через мессенджер Telegram и обеспечивать их доставку к указанному месту с использованием навигации через камеру.

# Архитектура программы

Используемые модули:

* Модуль для обработки изображений и распознавания объектов использует библиотеку OpenCV 4.7, которая работает с камерой.
* Модуль, отвечающий за управление роботом и навигацию к заданной цели.
* Модуль для отправки команд, который использует библиотеку mosquitto.
* Модуль Telegram, реализованный с помощью MQTT и TelegramAPI получает название напитка и отправляется в указанное место.

Диаграмма классов представлена на рисунке 1.

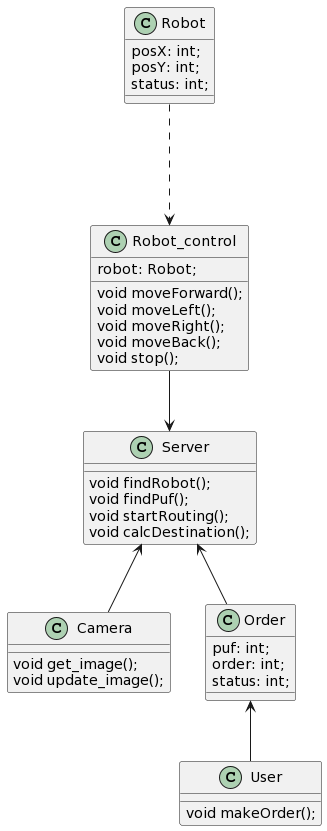


Рисунок 1 — UML-диаграмма классов

Система для управления роботом-доставщиком напитков содержит следующие классы:

1. Robot\_control: Класс, отвечающий за управление роботом. Содержит методы moveForward(), moveLeft(), moveRight(), moveBack(), stop() для управления движением робота.
2. Robot: Класс, представляющий робота. Содержит атрибуты posX и posY, обозначающие текущие координаты робота, и атрибут status, указывающий его текущий статус.
3. Camera: Класс, представляющий камеру. Содержит методы get\_image() и update\_image() для получения изображения с камеры и его обновления.
4. Order: Класс, представляющий заказ. Содержит атрибуты puf, order и status, отражающие информацию о заказе и его текущем статусе.
5. User: Класс, представляющий пользователя. Содержит метод makeOrder(), который позволяет пользователю создать заказ.
6. Server: Класс, представляющий сервер. Содержит методы findRobot(), findPuf(), startRouting() и calcDestination(), которые отвечают за поиск робота, поиск места доставки, запуск маршрутизации и расчет пункта назначения соответственно.

Связи между классами:

* Класс Robot связан с классом Robot\_control отношением композиции, что означает, что Robot\_control контролирует робота и может управлять его движением.
* Класс Server связан с классами Camera и Order, указывая на то, что сервер использует камеру для получения изображений и обрабатывает заказы.
* Класс Robot\_control связан с классом Server, что означает, что робот-доставщик может связываться с сервером для получения инструкций и отправки отчетов.
* Класс Order связан с классом User, указывая на то, что пользователь может создавать заказы.

Общее представление о работе классов на Sequence диаграмме — рисунке 2

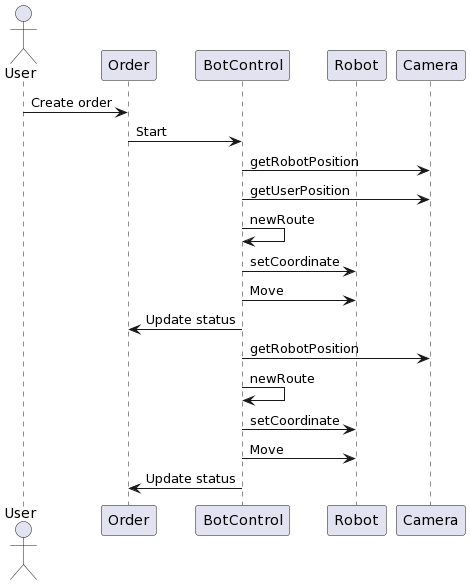


Рисунок 2 — Sequence диаграмма

Сценарий использования программы представлен на use-case диаграмме — рисунок 3.



Рисунок 3 — Use-case диаграмма

# Заключение

В ходе выполнения проекта по разработке робота-доставщика, мы получили практический опыт работы с различными инструментами и технологиями, такими как UML, CMake, Docker, MQTT, и телеграм-бот. Мы также изучили и успешно применили инструменты, такие как mosquitto (брокер MQTT-сообщений), OpenCV (библиотека компьютерного зрения) и PlantUML (ПО для создания UML-диаграмм). Это позволило нам разработать функциональное устройство для доставки напитков в коворкинге и получить ценный опыт и знания в данной области.

Ссылка на репозиторий с проектом: https://github.com/ksieuk/MAI\_Robot

# Источники

1. Библиотека OpenCV 4.7 – URL: https://github.com/opencv/opencv
2. Библиотека OpenSSL 3.1 – URL: https://github.com/openssl/openssl
3. Библиотека mosquitto 2.0.14 – URL: https://github.com/eclipse/mosquitto
4. Библиотека aiogram 2.25.1 – URL: https://github.com/aiogram/aiogram
5. Инструмент docker-compose 2.18.1 – URL: <https://github.com/docker/compose>
6. Проект MAI\_ROBOT – URL: https://github.com/ksieuk/MAI\_Robot