



**ПрофМир**

Институт непрерывного образования  
и профессионального развития

# ПРОГРАММА ПРОФЕССИОНАЛЬНОЙ ПЕРЕПОДГОТОВКИ

## **“Python-разработка для БПЛА: паттерны проектирования, API-интерфейсы и библиотеки для оптимизации решений”**

Преподаватель курса  
**Максимов Егор Васильевич**

Выполнил  
**Бубнович Антон Владимирович**

**№ БПЛА\_256-1**

**СОДЕЙСТВИЕ** | Федеральный  
**ЗАНЯТОСТИ** | проект

**Москва 2024 г.**

# Проект по теме

## «Создание полноценного Web-сайта и публикация его на GitHub»

### Содержание:

1. Описание проекта, его основные характеристики, цели и область применения
2. Применяемые паттерны проектирования, обоснование их выбора
3. API-интерфейсы для обеспечения взаимодействия программного решения с внешними системами или сервисами
4. Библиотеки для оптимизации решений в вашем проекте, объяснение их применения
5. Разработанная система тестирования ключевых компонентов проекта, включая отладку кода и обработку возможных ошибок
6. Анализ производительности проекта
7. Стратегии управления ресурсами и энергопотреблением для повышения эффективности работы программы
8. Документация к коду проекта
9. Демонстрация ключевой функциональности проекта



## Описание проекта, его основные характеристики, цели и область применения

### Актуальность темы:

В последние годы беспилотные летательные аппараты (БПЛА) стали неотъемлемой частью различных отраслей, таких как сельское хозяйство, логистика, мониторинг и спасательные операции. Программное обеспечение для управления этими устройствами играет ключевую роль в обеспечении их эффективности, безопасности и автономности. В рамках своей выпускной аттестационной работы я решил разработать и протестировать программное обеспечение для управления БПЛА, что позволило мне применить знания в области программирования, алгоритмов и системного проектирования.

## Описание проекта, его основные характеристики, цели и область применения

Целью моей работы было создание программного обеспечения для эффективного управления беспилотными летательными аппаратами (БПЛА). Оно должно обеспечивать автономное выполнение миссий БПЛА, стабильность полёта и минимизацию энергопотребления.

Для достижения этой цели я определил следующие задачи:

1. Спроектировать архитектуру системы управления БПЛА.
2. Реализовать основные модули программного обеспечения, включая модули управления полётом, обработки данных с сенсоров и принятия решений.
3. Интегрировать внешние API (Application Programming Interface) для расширения функциональности системы.
4. Оптимизировать и протестировать разработанное программное обеспечение, чтобы повысить его эффективность и надёжность.

## Описание проекта, его основные характеристики, цели и область применения

### Обзор архитектуры системы:

На этапе проектирования я разработал архитектуру системы управления БПЛА, которая состоит из трёх основных компонентов: сенсоров, исполнительных механизмов и блока управления.

Сенсоры собирают информацию о положении и движении летательного аппарата. Исполнительные механизмы отвечают за управление его движением. Блок управления анализирует полученные данные и принимает решения, руководствуясь заранее заданными алгоритмами.

Такая структура системы обеспечивает её стабильность и надёжность.

## **Описание проекта, его основные характеристики, цели и область применения**

### **Определение целей проекта:**

Основной целью проекта было создание системы, способной выполнять сложные задачи автономного полета, включая стабилизацию, навигацию и избегание препятствий. Для этого я разработал программное обеспечение, которое не только обрабатывает данные с сенсоров, но и адаптируется к изменениям окружающей среды.

### **Область применения:**

В процессе работы я ориентировался на использование БПЛА в различных областях, таких как сельское хозяйство для мониторинга полей, логистика для доставки грузов и поисково-спасательные операции. Разработанная система может быть адаптирована для решения конкретных задач в этих сферах.

## Применяемые паттерны проектирования, обоснование их выбора

### Применение паттерна MVC:

В процессе разработки программного обеспечения я использовал паттерн Model-View-Controller (MVC) для организации кода. Это позволило мне разделить логику приложения на три компонента: модель, представление и контроллер. Модель управляла данными о состоянии БПЛА, представление отвечало за визуализацию данных, а контроллер связывал модель и представление, обеспечивая логику управления.

## **Применяемые паттерны проектирования, обоснование их выбора**

### **Применение паттерна Observer:**

Для обработки данных с сенсоров я использовал паттерн Наблюдатель (Observer). Этот паттерн позволил мне настроить систему таким образом, что изменения в данных с сенсоров автоматически обновляли соответствующие компоненты системы управления. Это повысило оперативность и точность реакции системы на внешние изменения.

### **Применение паттерна State:**

Реализуя различные режимы полета, такие как взлет, круиз и посадка, я использовал паттерн Состояние (State). Каждый режим был представлен отдельным классом, что позволило легко управлять поведением БПЛА в зависимости от текущего состояния полета.



## **API-интерфейсы для обеспечения взаимодействия программного решения с внешними системами или сервисами**

### **Интеграция с внешними системами:**

В рамках проекта я интегрировал несколько внешних API, которые значительно расширили функциональность системы управления БПЛА. В частности, я использовал API для получения данных о погодных условиях, что позволило системе автоматически корректировать маршрут полета в зависимости от текущих метеоусловий.

## Библиотеки для оптимизации решений в вашем проекте, объяснение их применения

### Использование библиотеки для обработки сигналов:

Для улучшения точности данных, получаемых с сенсоров, я применил фильтр Калмана, реализованный с помощью специализированной библиотеки. Это позволило сгладить шумы и улучшить точность управления полетом.

### Оптимизация вычислительных процессов:

Для оптимизации работы системы я использовал библиотеку multiprocessing в Python, что позволило обрабатывать данные с сенсоров параллельно. Это решение значительно повысило скорость обработки данных и улучшило общую производительность системы.

## **Разработанная система тестирования ключевых компонентов проекта, включая отладку кода и обработку возможных ошибок**

### **Разработка системы тестирования:**

Для проверки корректности работы каждого модуля я разработал систему тестирования на основе библиотеки unittest. В рамках тестирования я проверял работу алгоритмов навигации, корректность обработки данных с сенсоров и взаимодействие между компонентами системы.

### **Обработка ошибок и отладка:**

Я реализовал механизмы обработки ошибок и отладки кода, что позволило оперативно выявлять и исправлять возникающие проблемы. Например, при потере связи с GPS системой были предусмотрены альтернативные методы навигации, что повысило надежность работы БПЛА.

## **Анализ производительности проекта**

### **Анализ производительности системы:**

Я провел оценку производительности системы управления БПЛА, измеряя время отклика на команды, точность навигации и скорость обработки данных с сенсоров. Для этого я использовал встроенные средства Python, такие как `timeit`, что позволило мне точно оценить эффективность системы.

### **Влияние оптимизаций:**

После внедрения оптимизаций проведен повторный анализ производительности и сравнились результаты с исходными данными. Было оценено, насколько сократилось время отклика системы после внедрения многопоточности.

# Стратегии управления ресурсами и энергопотреблением для повышения эффективности работы программы

## Оптимизация энергопотребления:

Для минимизации энергопотребления я реализовал несколько стратегий управления энергией. Например, я внедрил энергосберегающие режимы для сенсоров и двигателей, которые активировались в зависимости от текущих условий полета. Это позволило значительно увеличить время автономной работы БПЛА.

## Управление ресурсами:

Я разработал механизм управления ресурсами, который позволил эффективно использовать процессорное время и память. За счет оптимизации использования вычислительных ресурсов система смогла обрабатывать больше данных в реальном времени, не снижая при этом общей производительности.

## Документация к коду проекта

### **Структурированная документация:**

В ходе работы я создал подробное описание проекта. В нём я объяснил структуру системы, функции каждого модуля и класса, а также принципы работы алгоритмов. Это существенно облегчило процесс тестирования и отладки и подготовило основу для будущего развития проекта.

### **Комментарии и инструкции:**

Я снабдил код подробными комментариями, которые объясняют работу сложных алгоритмов и принятые решения. Кроме того, я подготовил инструкции по развёртыванию и настройке программного обеспечения. Это делает проект удобным для использования и адаптации.

## Демонстрация ключевой функциональности проекта





# ПрофМир

Институт непрерывного образования  
и профессионального развития

## Благодарю за внимание!

**СОДЕЙСТВИЕ** | Федеральный  
**ЗАНЯТОСТИ** | проект

[info@eduom.ru](mailto:info@eduom.ru)

