Самостоятельная работа. Модуль 1.
Швецов Леонид Сергеевич
«Разработка и отладка программного обеспечения для беспилотных летательных аппаратов (БПЛА). Методы тестирования и верификации».

Тема 8.3. Методы тестирования и верификации программного обеспечения БПЛА. Часть 2.

№ п/п	Наименование задания	Формат ответа
1	**Практическая работа по тестированию и верификации программного обеспечения для БПЛА:** В рамках данной работы необходимо применить методы тестирования и верификации, изученные на семинаре, для улучшения кода управления БПЛА.	1. Добавление функции посадки: - Загрузите исходный код по ссылке - https://gitlab.atp-fivt.org/kolpakovma/djicicd.git - Реализуйте метод land, который переводит дрон в режим посадки и контролирует успешное выполнение команды. - Напишите тесты для нового метода, используя pytest и unittest.mock.
		2. Расширение функциональности телеметрии: - Добавьте в метод get_telemetry обработку сообщений о скорости (VFR_HUD) и состоянии батареи (SYS_STATUS). - Обновите тесты, чтобы проверить получение и корректность новых данных.
		3. Интеграция с системой CI/CD: - Настройте автоматическое тестирование и анализ кода с использованием GitLab CI/CD, бесплатных раннеров GitLab. (необходимо изменить тег раннера). - Добавьте отчёты о покрытии кода и статическом анализе в пайплайн.
		4. Документирование кода: - Используйте инструмент Sphinx для генерации полной документации по проекту.
		5. Подготовка отчёта **Задание:** • **Подготовьте отчёт о проведённой работе, включающий описание реализованных функций, результаты тестирования, обнаруженные ошибки и их исправления.**

		• **Включите в отчёт выводы и рекомендации по дальнейшему улучшению кода.** **Выполнение:** **Описание реализованных функций** • **Метод land**: Реализован для безопасной посадки дрона, устанавливает режим LAND и контролирует выполнение команды. • **Расширение get_telemetry**: Добавлена обработка сообщений VFR_HUD и SYS_STATUS для получения данных о скорости, направлении и состоянии батареи. **Результаты тестирования** • Все тесты проходят успешно, включая новые тесты для метода land и расширенного get_telemetry. • Покрытие кода составляет более 90%. **Обнаруженные ошибки и их исправления** • **Ошибка в обработке исключений**: Обнаружено, что некоторые методы перехватывали слишком общие исключения (Exception). Исправлено путем использования более специфичных исключений (RuntimeError, ValueError). • **Недостаточное логирование**: Добавлены дополнительные сообщения в лог для улучшения отслеживания работы программы.
		6. Предоставьте обновлённый код модулей и тестов, соответствующий требованиям качества и безопасности. Код обновлён в соответствии с требованиями качества и безопасности: • **uav_control.py**: Добавлен метод land, обновлен get_telemetry, улучшена обработка исключений. • **test_uav_control.py**: Добавлены тесты для новых функций, покрытие кода увеличено. • **Документация**: Сгенерирована и доступна в репозитории.
		7. Перечислите используемые методы и инструменты тестирования, обоснуйте их выбор и эффективность. • **Методы тестирования**: ○ **Модульное тестирование**: Проверка отдельных функций и методов.

		- **Имитирование зависимостей**: Использование unittest.mock для мока объектов. - **Покрытие кода**: Оценка с помощью pytest-cov. • **Инструменты**: - **pytest**: Для организации и выполнения тестов. - **unittest.mock**: Для мока зависимостей и функций. - **pylint**: Для статического анализа кода. - **Sphinx**: Для генерации документации. - **GitLab CI/CD**: Для автоматизации тестирования и анализа кода. **Обоснование выбора и эффективность:** • **pytest**: Прост в использовании, позволяет писать понятные и гибкие тесты. • **unittest.mock**: Необходим для мока зависимостей, особенно при тестировании кода, взаимодействующего с внешними системами. • **pylint**: Помогает поддерживать качество кода, следовать стандартам. • **Sphinx**: Облегчает создание и поддержание документации. • **GitLab CI/CD**: Обеспечивает автоматизацию процессов тестирования и анализа, повышая надежность разработки.
		8. Загрузите свой код на GitHub или GitLab. Предоставьте публичный доступ для репозитория с вашим кодом. Несмотря на то, что мы используем инструменты CI/CD GitLab, я использую GitHub для всего проекта курса, поэтому размещу файлы там. В дальнейшем переделаю автоматизацию под GitHub Actions.
		9. Ссылку на репозиторий приложите в файл домашнего задания. https://github.com/lshvetsov/innopolis_uav/tree/master/module_8/8_3_2