

Самостоятельная работа. Модуль 8.

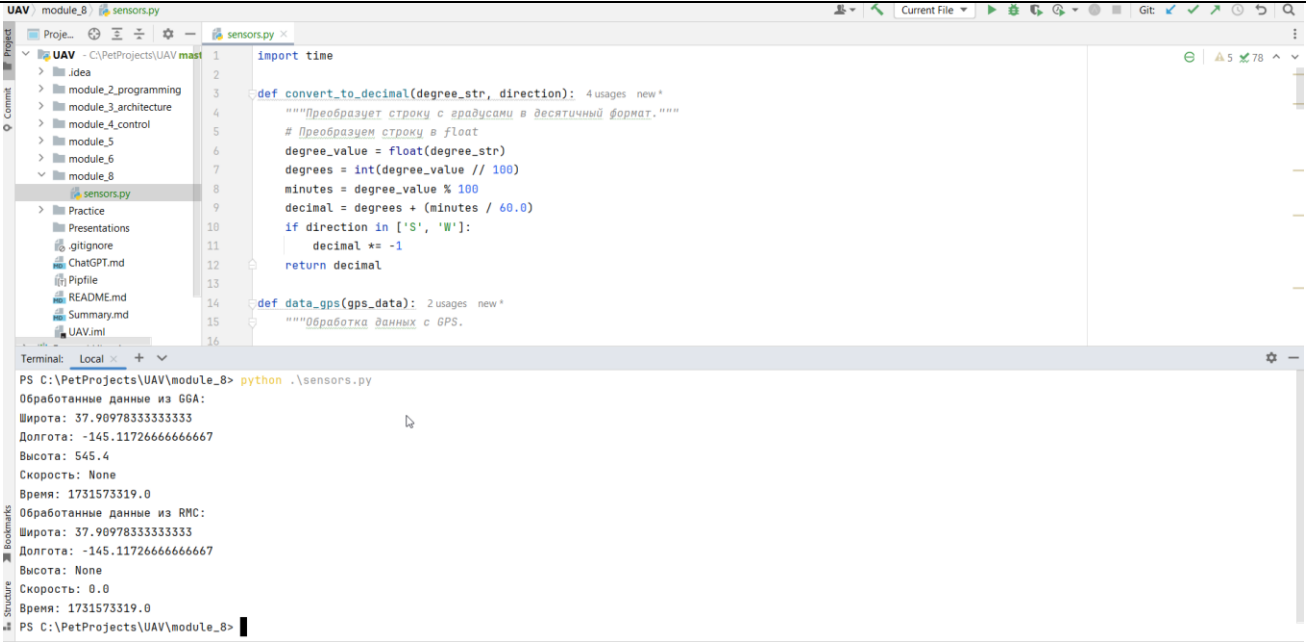
Швецов Леонид Сергеевич

«Разработка и отладка программного обеспечения».

Тема 8.2. Инструменты отладки и тестирования.

№ п/п	Наименование задания	Ответ
1	Изучение дополнительных материалов по отладке и тестированию.	1) Левашов Пётр. Python с нуля. – СПб.: Питер, 2024. – 448 с. 2) Donald Norris. Python for Microcontrollers. 2017. – 284 с. 3) Алгоритмы и структуры данных на Python. – СПб.: Питер. -368 с. 4) Балабанов П.В. Программирование беспилотного летательного аппарата мультироторного типа. 2023. – 86 с.
2	Выполнить дополнительное упражнение по отладке и тестированию.	2.1. Опишите основные цели отладки и тестирования при разработке программного обеспечения для БЛА. <ul style="list-style-type: none"><li>• <i>Обеспечение надежности:</i> ПО для БЛА должно работать стабильно и предсказуемо в различных условиях, поскольку любые сбои могут привести к потере аппарата или даже к опасным ситуациям.</li><li>• <i>Повышение безопасности:</i> Тестирование помогает обнаружить и устранить ошибки, которые могут привести к некорректному управлению БЛА или его системами, что критично для безопасности.</li><li>• <i>Улучшение производительности:</i> Отладка и тестирование помогают оптимизировать ПО, что важно для эффективного использования ресурсов БЛА, таких как время полета и энергопотребление.</li><li>• <i>Соответствие спецификациям и требованиям:</i> ПО для БЛА часто подвергается строгим требованиям и стандартам, которым оно должно соответствовать.</li></ul>

		<p>2.2. Опишите основные инструменты юнит-тестирования и интеграционного тестирования.</p> <ul style="list-style-type: none"> <li>• <i>Юнит-тестирование:</i> <ul style="list-style-type: none"> <li>• unittest: Встроенная библиотека Python, предоставляющая множество удобных функций для написания и выполнения тестов.</li> <li>• pytest: Популярная сторонняя библиотека для написания простых и масштабируемых тестов. Она поддерживает фикстуры, параметризацию и плагины.</li> <li>• nose2: Преемник nose, предоставляет расширенные возможности для тестирования, такие как автоматическое обнаружение тестов и плагины.</li> </ul> </li> <li>• <i>Интеграционное тестирование:</i> <ul style="list-style-type: none"> <li>• TestDoubles (mock, stub, fake): Инструменты, такие как unittest.mock, позволяют создавать моки и стабы, которые используются для имитации взаимодействий между компонентами системы.</li> <li>• Selenium: Инструмент для тестирования веб-приложений, который может быть использован для тестирования веб-интерфейсов управления БЛА.</li> <li>• Robot Framework: Фреймворк автоматизации, который поддерживает создание высокоуровневых сценариев тестирования, включая интеграционное и приемочное тестирование.</li> </ul> </li> </ul> <p>2.3. В выбранной среде разработки (IDE) подготовьте небольшой проект на примере анализа данных с сенсоров БЛА (например, акселерометр, гироскоп, барометр или GPS). Приложите скринкопию кода и файл с кодом.</p>
--	--	---

		 <p>sensors.py</p>
3	<p>Разработка небольшого проекта с использованием инструментов отладки и тестирования.</p>	<p>В выбранной интегрированной среде разработки (IDE) подготовьте небольшой проект на примере анализа данных с сенсоров БЛА (например, акселерометр, гироскоп, барометр или GPS). Приложите скринкопию проекта и архив проекта.</p> <p>Разработка проекта была проведена с применением юнит-тестирования, интеграционного тестирования и системного тестирования.</p> <p><b>1) Разработка проекта с применением юнит-тестирования.</b></p> <ul style="list-style-type: none"> <li>- был создан файл gps_module.py</li> <li>- был создан файл test_gps.py</li> <li>- в терминале интеграционной среды разработки Visual Studio Code была выполнена команда <code>python -m unittest test_gps.py</code> для запуска теста.</li> </ul>

		<p><b>2) Разработка проекта с применением интеграционного тестирования.</b></p> <ul style="list-style-type: none"> <li>- был создан файл <code>gps_module.py</code></li> <li>- был создан файл <code>integration_test_gps.py</code></li> <li>- в терминале интеграционной среды разработки Visual Studio Code была выполнена команда <code>python -m unittest integration_test_gps.py</code> для запуска теста.</li> </ul> <p><b>3) Разработка проекта с применением системного тестирования.</b></p> <ul style="list-style-type: none"> <li>- был создан файл <code>gps_module.py</code></li> <li>- был создан файл <code>system_test_gps.py</code></li> <li>- в терминале интеграционной среды разработки Visual Studio Code была выполнена команда <code>python -m unittest system_test_gps.py</code> для запуска теста.</li> </ul> <p>Отчёт о выполненном задании представлен далее при ответе на вопрос №4.</p>
4	Подготовка отчёта о выполненных заданиях.	<p>По итогам проведённого тестирования с применением юнит-теста, интеграционного теста и системного теста можно сделать несколько основных выводов относительно качества, надежности и функциональности разработанного небольшого проекта.</p> <p>1) Успешно был выполнен юнит-тест. Результат теста показал, что функция реализована правильно. Далее представлен результат теста.</p> <p>Целью юнит-теста является обеспечение корректного функционирования каждой отдельной единицы кода.</p> <p>2) Успешно был выполнен интеграционный тест. Результат теста показал, что функция, которая вызывается в ходе системы, возвращает ожидаемые результаты наборов данных. Далее представлен результат теста.</p> <p>3) Успешно был выполнен системный тест. Результат теста показал, как подготовленный алгоритм справляется с некорректными строками и крайними случаями (например, пустыми строками, недостаточным количеством полей или недостаточным количеством данных для анализа).</p>

Целью системного теста является проверка работы системы в целом с различными условиями, включая проверку корректного и некорректного ввода, можно добавлять дополнительные тестовые случаи, чтобы охватить больше сценариев.

```
PS C:\PetProjects\UAV\module_8> python -m unittest test_gps.py
```

```
..
```

```
-----  
Ran 2 tests in 0.004s
```

```
OK
```

```
PS C:\PetProjects\UAV\module_8> python -m unittest .\integration_test_gps.py
```

```
..
```

```
-----  
Ran 2 tests in 0.004s
```

```
OK
```

```
PS C:\PetProjects\UAV\module_8> python -m unittest .\system_test_gps.py
```

```
.Ошибка при обработке данных GPS: could not convert string to float: ''
```

```
...
```

```
-----  
Ran 4 tests in 0.005s
```

```
OK
```

```
PS C:\PetProjects\UAV\module_8>
```