МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего

образования

«Санкт-Петербургский государственный университет аэрокосмического приборостроения»

Кафедра информационно-сетевых технологий

Дисциплина «Интернет вещей»

Автор – ассистент Д.О. Шевяков

Лабораторная работа № 2

СОЗДАНИЕ ПРИЛОЖЕНИЯ ИНТЕРНЕТА ВЕЩЕЙ

И ВЕБ-ИНТЕРФЕЙСА

Цель работы – приобретение навыков создания и запуска приложения интернета вещей.

Методические указания

Для создания приложения интернета вещей необходим набор инструментов, который позволит реализовать веб-приложение, способное взаимодействовать с различным удалённым оборудованием. Flask — фреймворк для создания веб-приложений на языке программирования Python. Этот фреймворк обладает необходимым базовым функционалом, а все недостающие детали можно легко восполнить необходимыми модулями.

Для работы с данным фреймворком рекомендую установить pycharm professional, лицензия которой бесплатно предоставляется всем студентам аккредитованных учебных заведений, а также есть пробный период в 30 дней. В данной среде достаточно выбрать тип проекта flask и нужные модули будут автоматически установлены. Альтернативный способ установки Flask включает ручное создание виртуального окружения и запуск приложения в нем.

После создания проекта мы видим следующее содержание файла app.py. Это файл основного приложения, который включает все ссылки, по которым можно переходить. Запустив приложение и вбив в браузере адресс 127.0.0.1:5000 мы увидим созданную приложением веб-страницу.

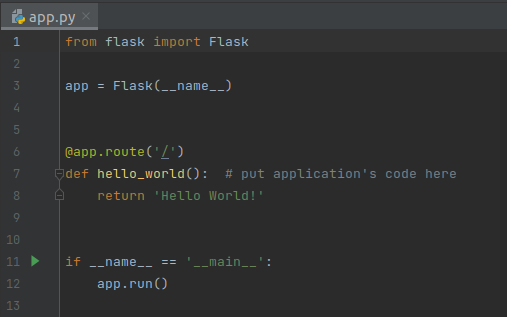


Рисунок 1 – Содержание файла app.py

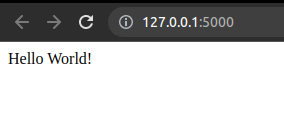


Рисунок 2 – Автоматически созданная страница

Далее нам необходимо создать в проекте новый файл, который будет содержать наши классы. Для создания класса в языке Python используется директива class. Затем в скобках указывается класс-родитель, если таковой имеется, по умолчанию python использует класс Base – базовый класс, являющийся родителем всех прочих.

Внутри каждого класса объявляется метод \_\_init\_\_. \_\_init\_\_ - это системный метод, который будет неявно создан в любом случае, когда класс будет объявлен, однако разработчик может его переопределить, как и любой другой системный метод. Этот метод исполняется в момент создания объекта класса и, как правило, в нём объявляются атрибуты класса. После всех системных методов идут пользовательские методы. Они объявляются, как отдельные функции, за исключение того, что у каждого метода есть входной параметр self, который указывает на текущий объект этого класса. Пример создания класса представлен ниже.

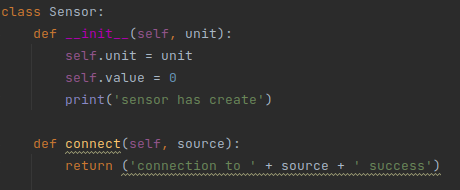


Рисунок 3 – Класс Sensor

Абстрактные классы в Python представляются модулем abc. Каждый абстрактный класс должен наследоваться от класса ABC этого модуля. Как известно, абстрактные классы не могут создавать объекты, а нужны только, чтобы от них наследовались другие классы, которые будут получить их методы и свойства. Перед каждым методом такого класса должен стоять декоратор «@abstractmethod». Однако, когда мы наследуем класс от абстрактного класса, необходимо будет вызвать в методе \_\_init\_\_ дочернего класса метод \_\_init\_\_ родительского абстрактного класса и добавить в метод вызова \_\_init\_\_ дочернего класса все атрибуты, которые были определены в родительском абстрактном классе.

Надо также заметить, что хоть абстрактный класс имеет метод \_\_init\_\_, этот метод может быть вызван только в дочернем классе и попытка создать объект абстрактного класса вызовет ошибку. Метод, описанный в абстрактном классе должен быть также реализован и в классе-потомке.

Ниже представлен абстрактный класс Thing и модифицированный класс Sensor.

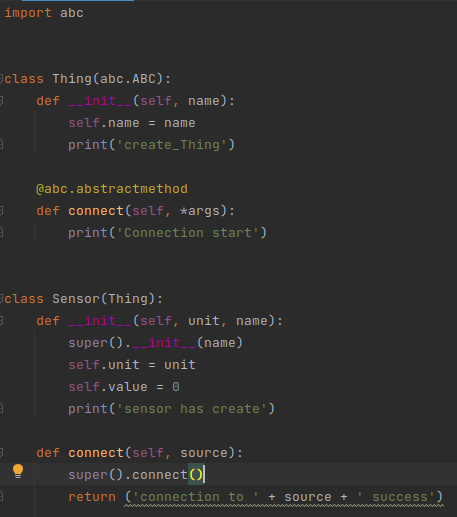


Рисунок 4 – класс Sensor, унаследованный от абстрактного класса Thing

Теперь, когда мы создали наши классы, можно приступить к реализации функционала наших классов. Вернёмся к файлу app.py, импортируем наш файл с классами и в функции hello\_world создадим объект нашего класса, затем обратимся к его атрибуту и вызовем его метод. Ниже представлен код и скриншоты лога нашего сервера и созданная веб-страница.

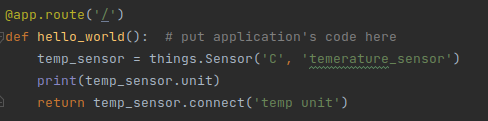


Рисунок 5 – Код функции hello\_world

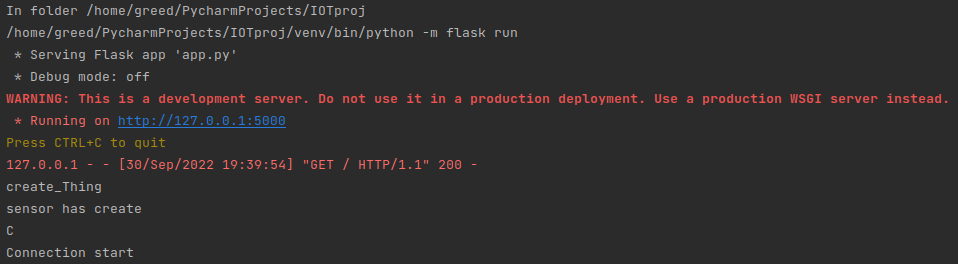


Рисунок 6 – Лог сообщений сервера

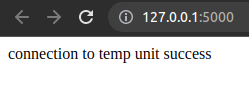


Рисунок 7 – Веб-страница

Задание

Создать приложение на Flask. Реализовать классы и их иерархию, согласно сданной первой лабораторной работе, показать работу методов этих классов. (Не нужно писать предполагаемый метод целиком, на данном этапе достаточно вывести в лог сервера сообщение о том, что метод был запущен)

Содержание отчёта:

1. Титульный лист
2. Задание
3. Код классов и приложения
4. Скриншоты
5. Выводы

Полезные ссылки:

1. <https://www.jetbrains.com/ru-ru/pycharm/>
2. <https://metanit.com/python/tutorial/7.1.php>
3. <https://docs-python.ru/tutorial/klassy-jazyke-python/abstraktnye-klassy/>
4. https://www.jetbrains.com/ru-ru/pycharm/