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 № 1

ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ ИНТЕРНЕТА ВЕЩЕЙ

И ВЕБ-ИНТЕРФЕЙСА

Цель работы – приобретение навыков проектирования приложения интернета вещей и интерфейса для отображения состояния системы.

Методические указания

В упрощённом и обобщённом виде любая система интернета вещей состоит из трёх слоёв, показанных на рисунке 1. (На самом деле слоёв больше, но в данном курсе они рассматриваться не будут)

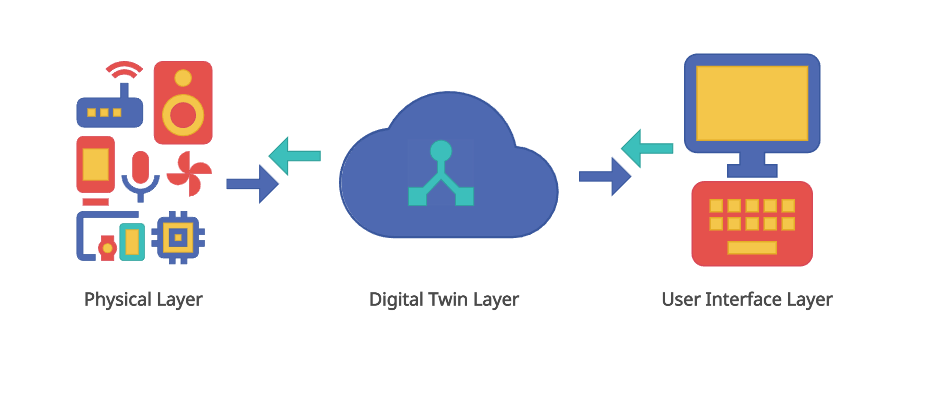


Рисунок 1 – Обобщённая схема системы интернета вещей

Остановимся подробнее на каждом слое:

Физический слой (Physical Layer) – слой реальных вещей, которые взаимодействуют с окружающим миром. На этом слое находятся все устройства, включенные в систему. Работа с этим слоем ограничена возможностями самих устройств, поэтому основная работа с данным слоем заключается в подборе этих самых устройств под ваши требования.

Слой цифровых двойников (Digital Twin Layer) – слой, как правило, расположенный на выделенном сервере и представляющий собой набор цифровых двойников реальных устройств. Данный слой отвечает за управление, мониторинг, обработку и обмен информацией между вещами системы. Цифровые двойники позволяют без особых трудностей обрабатывать информацию, поступающую с реальных устройств, поскольку данный слой ограничен лишь вычислительными мощностями сервера.

Слой интерфейсов пользователя (User Interface Layer) – слой, предназначенный для отображения информации с устройств и управления ими. Может быть реализован самыми разными способами, например, веб-интерфейс.

Цифровые двойники в системе, как правило представлены набором объектов классов, которые заранее спроектированы. Так, например, можно создать класс для датчика и внести в него все необходимые свойства и методы. Когда нам понадобится включить новый датчик в системы, мы просто можем создать новый объект этого класса.

Таким образом, проектирование слоя цифровых двойников сводится к созданию максимально продуманной диаграммы классов, которая будет включать в себя все необходимые свойства и методы для функционирования системы.

Диаграмма классов UML представляет собой набор таблиц различного содержания, которые описывают классы, их атрибуты, методы и отношения между объектами. Основным объектом этой диаграммы, как не сложно догадаться, является Класс. Класс представляет собой шаблон для создания объектов, обеспечивающий начальные значения состояний: инициализацию полей-переменных и реализацию поведения полей и методов. На рисунке 2 представлен пример класса.

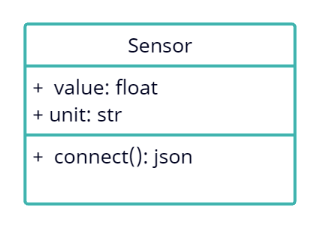


Рисунок 2 – Класс «Sensor»

Между классами могут (и должны) быть отношения. Отношения бывают разные, но наиболее часто использующиеся из них – это ассоциации и наследование.

Ассоциации показывают связь объектов системы и показываются прямой линией, соединяющей два класса с указанием смыла этого взаимодействия. Также можно указать число объектов, которые могу учувствовать в отношениях, оно записывается в виде интервала или точного числа, которое обозначает количество объектов с той или иной стороны. 1..4 – от одного до четырёх объектов. \* - сколько угодно объектов. 1..\* - Сколько угодно, но не менее одного объекта. На рисунке 3 показан пример ассоциации. Из этого рисунка видно, что множество объектов класса Sensor могут отправлять данные одному объекту класса MainControlUnit.

Наследование отображает отношения между родительским и дочерним классами. Обозначится полой стрелкой, направленной на родительский класс. Классы могут также наследоваться от абстрактного класса. Абстрактный класс – это класс, который лишь описывает некую структуру и набор методов, но не предполагает возможность создания объектов. Такой класс представляет собой некую абстрактную модель, имеющую некоторый объём, но не имеющую содержания. Такой класс может быть полезен при проектировании для создания общего для нескольких классов набора свойств и методов. На Диаграмме такой класс отличается от обычного лишь тем, что его название указано курсивом. На рисунке 4 представлен пример наследования от абстрактного класса. В данном случае объект класса Sensor будет иметь не только свойства и методы своего класса, но и также свойства и методы класса Thing.

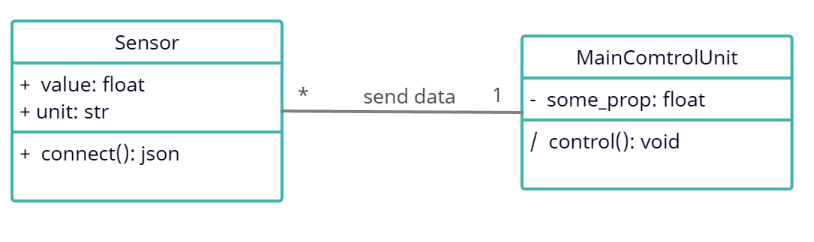


Рисунок 3 – Пример ассоциации

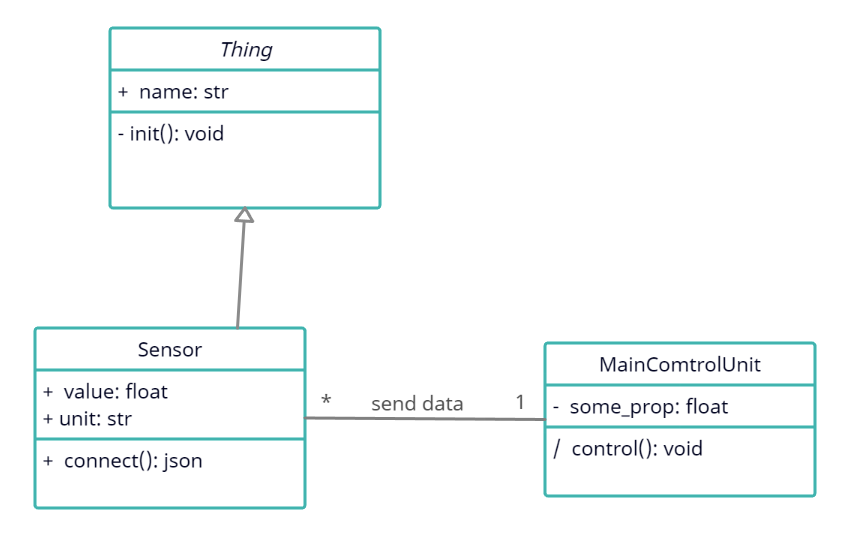


Рисунок 4 – Наследование от абстрактного класса

Проектирование интерфейса – скорее творческая задача, чем точная наука. Интерфейс может быть реализован в виде схематического рисунка с указанием областей, в которых будет располагаться нужная информация. В рамках данной лабораторной работы будет оцениваться практичность и удобство интерфейса, а не его красота.

Задание

1. В рамках предложенной (или своей) темы спроектируйте систему, состоящую из 4-5 уникальных классов. Подумайте имеют ли классы общие методы и свойства, если таковые есть, выделите их в абстрактный класс. Обозначьте связь элементов системы (Подсказка: не лишним будет создать отдельный объект, который не имеет физического воплощения, а отвечает только за обработку и передачу данных между другими объектами, если это необходимо, в моём примере – это MainControlUnit). Подумайте какие данные в системе будет хранить база данных, обозначьте её отдельной сущностью, связями обозначьте какие данные и с каких вещей она должна хранить. Поскольку следующие лабораторные подразумевают написание кода на Python, в котором модификаторы доступа носят рекомендательный характер, ими можно пренебречь. (Это я про «+», «-» и «/» на диаграмме)

2. В рамках предложенной (или своей) темы спроектируйте, как минимум два интерфейса. Два интерфейса могут быть нужны, например, для пользователя и администратора. На примере умного дома это может быть интерфейс ребёнка, который только отображает некоторые данные и позволяет управлять некоторыми из вещей и интерфейс родителя, которые может позволять более глубоко настраивать управление домом. Подумайте какие данные лучше скрыть от одного пользователя, какие лучше показать, как лучше их отобразить.

Варианты заданий:

1. Умный дом
2. Умный гараж
3. Умная теплица
4. Умный ресторан
5. Умный склад
6. Умная улица (упрощённая версия умного города)
7. Умная производственная ячейка
8. Умная больничная палата
9. Умное игровое поле (для вашей любимой спортивной игры)
10. Умная система слежения за состоянием человека (Всё, чем мы пользуемся в повседневной жизни)

Содержание отчёта:

1. Титульный лист
2. Задание
3. Диаграмма классов с описанием всех классов
4. Скриншоты интерфейсов с пояснениями
5. Выводы

Полезные ссылки:

1. <https://habr.com/ru/post/511798/>

2. https://ux.pub/editorial/10-pravil-proiektirovaniia-intierfieisov-kotoryie-nielzia-narushat-21m6