TOMFramework Project

Class Design Document

Version: 0.1 (Draft version)

Authors:

|  |  |  |
| --- | --- | --- |
| No. | Student ID | Student name |
| 19 | B2110132 | Lưu Trí Nguyên |

*The purpose of this document is the draft version of class design that provides you with a guideline for detailing class design and writing the class diagram description for your solution based on design Pattern approach.*

*Points to remember:*

* *Content is important, not the volume.* ***Another team should be able to design in more detail from this document.***
* *Pay attention to overall in description.*
* *Completeness and consistency will be rewarded.*
* SINH VIÊN ĐƯỢC PHÉP XEM TÀI LIỆU
* SINH VIÊN KHÔNG ĐƯỢC SAO CHÉP HAY PHỔ BIẾN TÀI LIỆU TRONG PHÒNG THI DƯỚI MỌI HÌNH THỨC.

Contents

[1. Name: Enviromental Sensor Data Accquistion 4](#_Toc161236640)

[2. Problem 4](#_Toc161236641)

[3. Solution 4](#_Toc161236642)

[3.1. Class diagram 4](#_Toc161236643)

[3.2. Class Descriptions 4](#_Toc161236644)

[3.2.1. Class: Instrument 5](#_Toc161236645)

[3.2.2. Class Handy 7](#_Toc161236646)

[3.2.3. Class Station 7](#_Toc161236647)

[4. Consequence 7](#_Toc161236648)

[5. Implementation (code examples for Receive, HTTP and MQTT only) 8](#_Toc161236649)

## Name: Enviromental Sensor Data Accquistion

## Problem

* Hệ thống cần thu thập dữ liệu môi trường từ các dụng cụ đo đạc
* Có nhiều loại cảm biến và thiết bị truyền tin khác nhau
* Cần có một thiết kế phần mềm linh hoạt để hỗ trợ nhiều loại dụng cụ và thiết bị

## Solution

* Sử dụng Strategy pattern và template method pattern để thiết kế phần mềm
* Strategy pattern được sử dụng để thay đổi hành vi của các cảm biến và thiết bị truyền tin
* Template method pattern được sử dụng để định nghĩa một khung chung cho việc thu thập dữ liệu từ các dụng cụ.

## Class diagram



## Class Descriptions

***Pay attention: only describe Instrument and its subclass (Handy and Station)***

### Class: Instrument

* Purpose: Lớp cơ bản cho các dụng cụ đo đạc
* Constraints: None
* Persistent: No (created at system initialization from other available data)

Attribute Descriptions

1. Attribute: instrumentCategory

Type: int

Description: định danh loại dụng cụ

Constraints: no

1. Attribute: instrumentID

Type: int

Description: định danh dụng cụ

Constraints: no

1. Attribute: sensors

Type: List<Sensor>

Description: Danh sách các cảm biến được gắn vào công cụ

Constraints: 0 <= sensors.length <= 4

Method Descriptions

1. Method: getTime()

Return Type: String

Parameters:None

Return value:chuỗi kí tự thời gian với định dạng YYYY-MM-DD-hh:mm-ss

Pre-condition:None

Post-condition: Không thay đổi trạng thái của công cụ

Attributes read/used:None

Methods called:None

Processing logic: Sử dụng lớp Date để lấy thời gian hiện tại. Định dạng thời gian theo định dạng YYYY-MM-DD-hh:mm-ss. Trả về chuỗi kí tự chứa thời gian

Test case 1: Gọi phương thức getTime(). Giá trị trả về dự kiến là: chuỗi kí tự thời gian hiện tại theo định dạng YYYY-MM-DD-hh:mm-ss

1. Method: readSensor()

Return Type: String

Parameters: None

Return value: Chuỗi kí tự dạng JSON

Pre-condition: Có ít nhất một cảm biến được gắn vào dụng cụ

Post-condition: Không thay đổi trạng thái của dụng cụ

Attributes read/used: sensors

Methods called: getIndicator()

Processing logic: Duyệt qua danh sách các cảm biến được gắn vào dụng cụ. Gọi phương thức getIndicator() của mỗi cảm biến để lấy giá trị đo được. Lưu trữ giá trị đo được vào một cấu trúc dữ liệu. Chuyển đổi cấu trúc dữ liệu thành chuỗi JSON. Trả về chuỗi JSON chứa giá trị đọc được từ cảm biến

Test case 1:Gọi phương thức readSenssor() cho dụng có một cảm biến DO. Giá trị trả về dự kiến: Chuỗi JSON chứa giá trị nồng độ DO

Test case 2: Gọi phương thức readSensor() cho dụng cụ có hai cảm biến DO và PH. Giá trị trả về dự kiến: Chuỗi JSON chứa giá trị nồng độ DO và giá trị độ PH.

1. Method: saveData()

Return Type: void

Parameters: None

Return value: None

Pre-condition: Dữ liệu cảm biến đã được đọc

Post-condition: Dữ liệu cảm biến được lưu vào thẻ nhớ

Attributes read/used: sensors

Methods called: readSensor(), writeToFile() (phương thức ghi dữ liệu vào File).

Processing logic: Gọi phương thức readSensor() để lấy giá trị đo được từ cảm biến. Chuyển đổi giá trị đo được thành định dạng phù hợp để lưu trữ. Gọi phương thức writeToFile() để ghi dữ liệu vào thẻ nhớ.

Test case 1: Gọi phương thức saveData() sau khi đã đọc dữ liệu từ cảm biến. Kết quả dự kiến: Dữ liệu cảm biến được lưu vào file trên thẻ nhớ.

1. Methods: sendData()

Return Type: int

Parameters:

* + severID: Chuỗi xác định địa chỉ
  + account: Tài khoản kết nối
  + password: mật khẩu
  + strData: giá trị truyền
  + strTime: chuỗi thời gian

Return value: int (0 thành công, khác 0: lỗi)

Pre-condition: Dữ liệu cảm biến đã được đọc

Post-condition: Dữ liệu cảm biến được gửi đến địa chỉ Data server

Attributes read/used: no

Methods called: readSensor(), send() của lớp TransEquipment

Processing logic: Gọi phương thức readSensor() để lấy giá trị đo được từ các cảm biến. Chuyển đổi giá trị đo được thành định dạng phù hợp để truyền. Gọi phương thức send() của lớp TransEquipment để gửi dữ liệu đến Data Server. Xử lý kết quả trả về từ phương thức send(). Nếu gửi thành công trả về 0 , nếu gửi thất bại trả về mã lỗi phù hợp.

.

Test case 1: Gọi phương thức sendData() với kết nối internet ổn định. Kết quả dự kiến: Dữ liệu cảm biến được gửi thành công đến Data Server và phương thức trả về 0.

Test case 2: Gọi phương thức sendData() với kết nối internet không ổn định. Kết quả dự kiến: Dữ liệu cảm biến được gửi thành công đến Data Server và phương thức trả về mã lỗi phù hợp.

1. Methods: loop()

Return Type: void

Parameters: None

Return value: None

Pre-condition: None

Post-condition: Lặp lại các bước: đọc thời gian, đọc giá trị cảm biến, lưu dữ liệu, gửi dữ liệu

Attributes read/used: no

Methods called: getTime(), readSensor(), saveData(), sendData()

Processing logic: Lặp lại vô hạn:

Gọi phương thức getTime() để lấy thời gian hiện tại.

Gọi phương thức readSensor() để lấy giá trị đo được từ cảm biến

Gọi phương thức saveData() để lưu dữ liệu cảm biến vào thẻ nhớ

Gọi phương thức sendData() để gửi dữ liệu đến Data Server

Test case 1: Gọi phương thức loop(). Kết quả dự kiến: Lặp lại các bước : đọc thời gian, đọc giá trị cảm biến, lưu dữ liệu, gửi dữ liệu.

### Class Handy

* Purpose: Lớp con của Instrument cho dụng cụ cầm tay
* Constraints: Chỉ gán được 1 cảm biến
* Persistent:No (created at system initialization from other available data)

Attribute Descriptions: None

Method Descriptions: None

### Class Station

* Purpose: Lớp con của Instrument cho dụng cụ cố định
* Constraints: Có thể gán từ 1 đến 4 cảm biến
* Persistent:No (created at system initialization from other available data)

Attribute Descriptions: None

Method Descript: None

## Consequence

* Lợi ích:
  + Dễ dàng thêm hoặc bớt các cảm biến (Sensor) mà không cần sửa đổi mã hệ thống
  + Dễ mở rộng hệ thống để đo lường các thông số môi trường mới
  + Cho phép tái sử dụng các thành phần Sensor và TransEquipment trong các hệ thống khác
* Hạn chế:
  + Việc thiết kế và triển khai hệ thống có thể phức tạp hơn so với các hệ thống truyền thống
  + Khó khăn trong việc quản lý và theo dõi các thành phần

## Implementation (code examples for Receive, HTTP and MQTT only)