**BÁO CÁO ĐỒ ÁN**

**MẪU THIẾT KẾ HƯỚNG ĐỐI TƯỢNG**

VALIDATION FRAMEWORK

**Nhóm sinh viên:**

1612689 – Trương Phạm Nhật Tiến

1612705 – Lê Thanh Thành Toại

1612714 – Nguyễn Hồng Tới

1612726 – Nguyễn Minh Trí

**Giáo viên thực hành:**

Mai Anh Tuấn

**Năm học: 2019 - 2020**

C:\Users\tdqua_000\Dropbox\SS-Slides\DeCuong-CDIO\Template CDIO v4.2\Templates\Hinh anh\LogoTruong.png

Bộ môn Công nghệ phần mềm

Khoa Công nghệ thông tin

Đại học Khoa học tự nhiên TP HCM

1. **Sơ đồ lớp**

**MỤC LỤC**

[**I.** **Sơ đồ lớp toàn bộ Framework** 3](#_Toc29115063)

[**1.** **Sơ đồ lớp** 3](#_Toc29115064)

[**2.** **Ý nghĩa của từng lớp** 3](#_Toc29115065)

[**3.** **Các mẫu sử dụng trong đồ án** 4](#_Toc29115066)

[**a.** **Singleton** 4](#_Toc29115067)

[**b.** **Factory** 6](#_Toc29115068)

[**c.** **Strategy** 7](#_Toc29115069)

[**d.** **Decorator** 8](#_Toc29115070)

[**e.** **Bridge** 9](#_Toc29115071)

[**II.** **Danh sách các tính năng** 10](#_Toc29115072)

1. **Sơ đồ lớp toàn bộ Framework**
2. **Sơ đồ lớp**

A close up of a map

Description automatically generated

1. **Ý nghĩa của từng lớp**

* **LanguageNotification:** Đây là lớp interface chứa hàm trả về kiểu String là nội dung thông báo lỗi để hiển thị cho người dùng.
* Ý nghĩa của lớp là tạo một giao diện chung cho cách hiển thị thông báo lỗi theo ngôn ngữ, tùy vào người dùng lựa chọn mà có thể linh động hiển thị theo ý của người dùng.
* **English:** Đây là lớp con của **LanguageNotification**, trả về nội dung thông báo lỗi hiển thị bằng Tiếng Anh.
* **Vietnamese**: Đây là lớp con của **LanguageNotification**, trả về nội dung thông báo lỗi hiển thị bằng Tiếng Việt.
* **LanguageFac:** một class chịu tránh nhiệm khởi tạo các đối tượng sub class dựa theo tham số đầu vào. Lưu ý: lớp này là [Singleton](https://gpcoder.com/4190-huong-dan-java-design-pattern-singleton/)hoặc cung cấp một public static method cho việc truy xuất và khởi tạo đối tượng. Factory class sử dụng if-else hoặc switch-case để xác định class con đầu ra.
* **ShowBehavior:** là một interface là giao diện chung cho cách hiển thị thông điệp ra màn hình với hàm show().
* **MessBox:** với phương thức show()được override từ **ShowBehavior** có nhiệm vụ hiển thị tất cả error của tất cả các trường đang kiểm tra bằng cách MessageBox.Show().
* **Popup:** với phương thức show()được override từ **ShowBehavior** có nhiệm vụ hiển thị tất cả error của tất cả các trường đang kiểm tra bằng cách new Windows.
* **ListValidator:** là lớp chứa các **Validator** và có phương thức show() để có thể show tất cả Message ứng với từng kiểu **ShowBehavior**.
* **Validator:** lớp này là abtract class - bộ kiểm trả dùng để kiểm tra lỗi, lớp này có phương thức check() để kiểm tra xem data có hợp lệ hay không.
* **ValidatorExtension:** là một abstract class dùng để duy trì một tham chiếu đối tượng Validator và đồng thời cài đặt các phương thức của **Validator**.
* **IsEmailExtension, PhoneViettel:** là lớp thực hiện các phương thức của **ValidatorExtension**, có thể cài đặt thêm các tính năng mới của chính nó**.**
* **IsEmail, IsEmpty, Phone, CreditCard, IsPassword:** là lớp con của Validator, nó sẽ thực hiện việc kiểm tra xem dữ liệu truyền vào có hợp lệ hay không**.**
* **ValidateInterger, ValidateString, ValidateRegex:** các lớp hỗ trợ validate cho các kiểu dữ liệu căn bản**.**

1. **Các mẫu sử dụng trong đồ án**
2. **Singleton**
   * + Sơ đồ lớp

A picture containing indoor

Description automatically generated

* + - Đoạn code sử dụng

A screenshot of a cell phone

Description automatically generated

* + - Ý nghĩa: Trong giới hạn project thì ta chỉ cần một factory để tạo Language, do đó để tránh tình trạng lãng phí tài nguyên không cần thiết thì áp dụng mẫu Singleton vào là cần thiết.

1. **Factory**
   * + Sơ đồ lớp

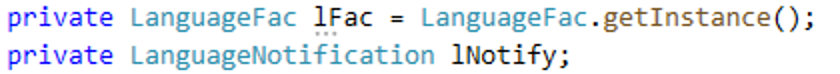
A close up of a map

Description automatically generated

* + - Đoạn code sử dụng:

A screenshot of a social media post

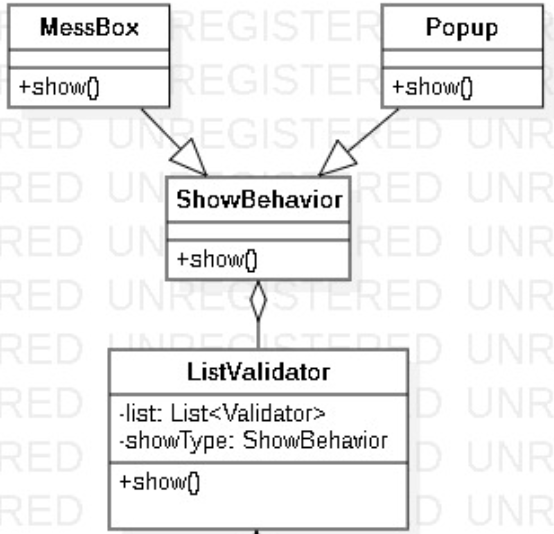
Description automatically generated

A screenshot of a cell phone

Description automatically generated

* + - Ý nghĩa: che giấu sự khởi tạo của đối tượng, người dùng sẽ không thấy cách mà đối tượng tạo ra.

1. **Strategy**
   * Sơ đồ lớp



* + - Đoạn code sử dụng

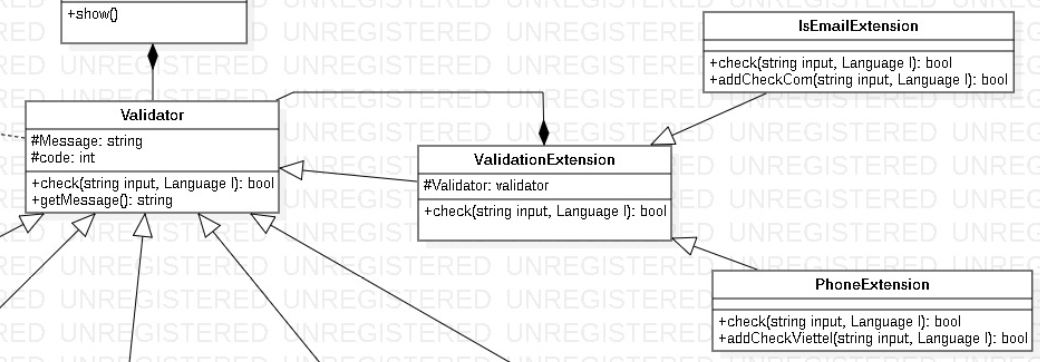
A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

* + - Ý nghĩa: tách phần hiển thị thông điệp ra riêng với Validator, do đó ta có thể thay cách hiển thị trực tiếp khi đang chạy chương trình.

1. **Decorator**
   * Sơ đồ lớp



* Đoạn code mẫu:

A screenshot of a social media post

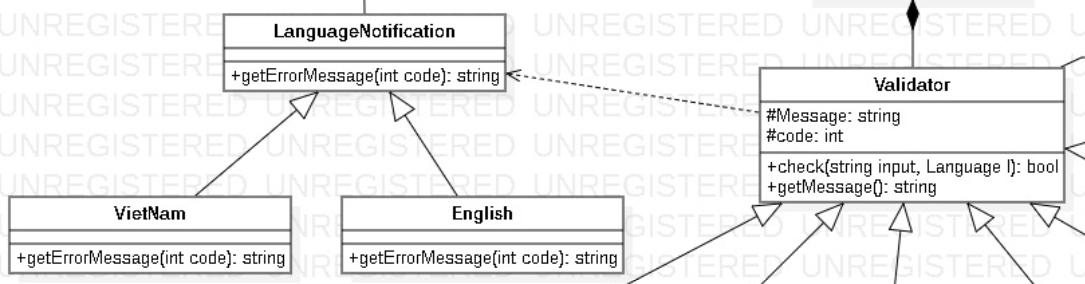
Description automatically generated

* Cách sử dụng: A screenshot of a cell phone

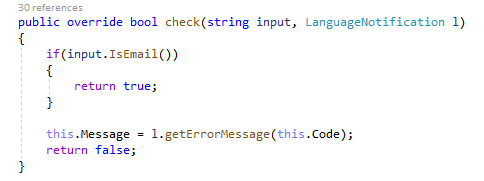
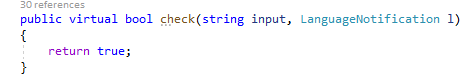
  Description automatically generated
* Ý nghĩa: có thể mở rộng validator có thể kiểm tra thêm một số thứ ngoài bản thân nó hoặc có thể thêm validator để kiểm tra.

1. **Bridge**

* Sơ đồ lớp:



* Cách sử dụng:



* Ý nghĩa: tách rời phần LanguageNotification ra khỏi Validator để hai bộ phận có thể thay đổi độc lập với nhau. Dễ dàng mở rộng khi thêm ngôn ngữ mới hay validate mới mà không ảnh hưởng đến những nơi sử dụng lớp ban đầu.

1. **Danh sách các tính năng**

|  |  |
| --- | --- |
| Tên tính năng | Mức độ hoàn thành |
| Cơ chế thông báo khi data không valid   * Tổng hợp các thông báo * Có các cách khác nhau thể hiện thông báo trên giao diện | **100%** |
| Thao tác thiết lập valid bằng code   * Tự kiểm tra với code thêm vào | **100%** |
| Thao tác thiết lập valid tự động thông qua khai báo ràng buộc dữ liệu | **100%** |
| Có thể kết hợp các validation với nhau cho cùng kiểu dữ liệu | **100%** |
| Hỗ trợ kết hợp regular expression | **100%** |
| Cho phép tạo custom validation | **100%** |