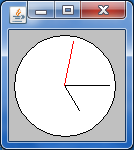
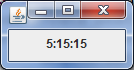
**Bài tập ôn**

1. **Kiểm tra hợp lệ dữ liệu của thành phần**

Một thành phần ***Component*** chứa nội dung cần kiểm tra tính hợp lệ của dữ liệu. Có các loại thành phần ***TextComponent*** chứa nội dung văn bản và ***NumberComponent*** chứa nội dung số. Có nhiều loại kiểm tra sự hợp lệ (***Validation***) của dữ liệu: kiểm tra trên thành phần văn bản có là dạng số hay không (dữ liệu số chứa toàn số và có nhiều nhất 1 dấu chấm thập phân), kiểm tra có là email hay không (chứa một ký tự @), kiểm tra có là số điện thoại hay không (có 10 ký tự bắt đầu bằng số 08), chiều dài lớn nhất và nhỏ nhất. Kiểm tra trên dữ liệu số có giá trị lớn nhất, giá trị nhỏ nhất. Mỗi thành phần có thể thực hiện nhiều kiểm tra đồng thời. Một kiểm tra trên một thành phần gọi là thành công nếu mọi kiểm tra đều thành công.

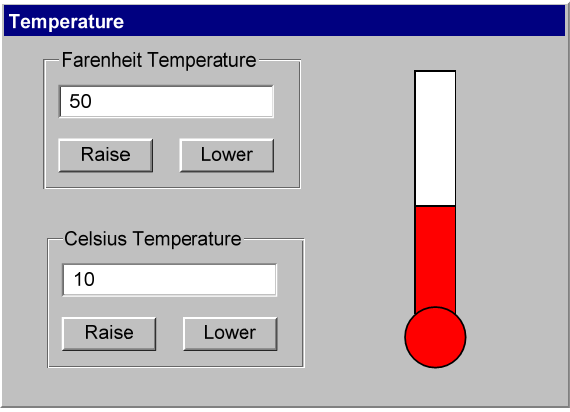
1. Thiết kế và cài đặt thành phần ***Component***, ***TextComponent***, ***NumberComponent*** và các kiểu kiểm tra sự hợp lệ của dữ liệu trên các thành phần và có thể dễ dàng thêm các thành phần và các kiểu kiểm tra khác.
2. Cài đặt các kiểu kiểm tra sự hợp lệ của thành phần văn bản có dạng số, email, số điện thoại như trên và các kiểu kiểm tra sự hợp lệ của thành phần số có giá trị lớn nhất, giá trị nhỏ nhất.
3. **Đồng hồ**

Viết chương trình hiển thị 2 đồng hồ loại digital và analog cho thời gian hệ thống.

Thêm các hiển đồng hồ khác hiển thị thời gian của hệ thống như đồng hồ kép, đồng hồ cho một thành phố nào đó.

1. **Chuyển đổi nhiệt độ**

Bạn phải thiết kế ứng dụng cho phép người dùng hiệu chỉnh giá trị nhiệt độ dùng các nút **Raise** và **Lower** trong các phần **Celsius** hay **Fahrenheit** của giao diện bên. Khi giá trị trong một hộp văn bản thay đổi, các giá trị trong các thang đo khác sẽ được cập nhật theo. Thanh nhiệt độ vẽ bên phải cũng cập nhật tương ứng.

* Dùng mẫu **Observer** để thiết kế các thành phần giao diện được cập nhật tương ứng khi dữ liệu trong model được thay đổi bởi một một thành phần.

1. **Đặt tour du lịch**

Một công ty du lịch XYZ cần quản lý các thông tin về các tour du lịch mà công ty có thể tổ chức, thông tin **Tour** bao gồm: tên tour (**name**), miêu tả chương trình tour (**description**), số ngày đi (**days**), giá tiền (**basePrice**). Một tour có thể đi trong nước (**DomesticTour**) hay tour nước ngoài (**OverseasTour**), nếu là tour nước ngoài thì thì có thêm thông tin về quốc gia đến (**country**), và tiền phụ thu visa (**visaSurtax**).

1. Thiết kế và cài đặt các lớp để biểu diễn **Tour** du lịch và tính giá cho các tour như sau

Giá tour (**price**) = Giá cơ bản tour, nếu là tour trong nước.

= Giá cơ bản tour + Phụ thu Visa, nếu là tour nước ngoài.

Tạo test với các dữ liệu tour như sau:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **name** | **description** | **days** | **basePrice** | **country** | **visaSurtax** |
| DLat | Đà lạt | 3 | 550,000 |  |  |
| NTrang | Nha trang | 4 | 900,000 |  |  |
| PQ | Phú Quốc | 4 | 1,250,000 |  |  |
| Thai | HCM - Bangkok - Pattaya - Koh Island | 5 | 3,600,000 | Thailand | 500,000 |

1. Người ta muốn thêm yêu cầu mới như giảm giá tour, nâng cấp tour thành 5 sao, tour kết hợp thăm thân nhân, hoặc thêm thông tin và ứng xử khác cho các tour có sẵn mà không hiệu chỉnh thiết kế tour trên.

* Áp dụng mẫu **Decorator** vào thiết kế để có thể dễ dàng thêm thông tin và ứng xử cho các tour có sẵn. Thiết kế và cài đặt tour giảm giá (**SaleTour**) bao gồm thêm thông tin số tiền giảm giá (**saleAmount**), và tour kết hợp thăm thân nhân (**VisitFamilyTour**).
* Tính lại giá cơ bản cho các tour như sau:

Giá cơ bản tour giảm giá = Giá cơ bản tour gốc – Số tiền giảm giá

Giá cơ bản tour kết hợp thăm thân nhân = Giá cơ bản tour gốc \* 1.05

* Cài đặt lại phương thức **getDescription()** cho các tour bao gồm thêm thông tin số tiền giảm giá hoặc thông tin tour thăm thân nhân tương ứng.

1. **Chuyển đổi tiền tệ**

Thiết kế một chương trình chuyển đổi tiền tệ. Chương trình có thể làm việc với nhiều loại đơn vị tiền tệ (**CurrencyUnit**), ví dụ USD, đồng VN, Yên Nhật, Euro, Bảng Anh… Mỗi loại đơn vị tiền tệ có một ký hiệu tiền tệ(**symbol**), ví dụ USD là "$", VNĐ là "VND", Yên Nhật là "JPY", Euro là "EUR", Bảng Anh là "GBP", … Một lượng tiền (**CurrencyMeasure**) của một đơn vị tiền tệ có thể chuyển đổi thành một lượng tiền của một đơn vị tiền tệ khác theo tỉ giá chuyển đổi. Thông tin biểu diễn lượng tiền gồm giá trị (**value**) và đơn vị tiền tệ (**unit**).

1. Thiết kế các lớp biểu diễn các đơn vị tiền tệ và lượng tiền của một đơn vị tiền tệ. (1.5đ)
2. Để chuyển đổi lượng tiền **MA** của đơn vị tiền tệ **A** thành lượng tiền **MB** của đơn vị tiền tệ **B** người ta có thể thực hiện 2 bước sau:

* Chuyển đổi lượng tiền **MA** của đơn vị tiền tệ **A** thành lượng tiền **MS** của đơn vị tiền tệ cơ sở (**MS = MA / rA**, với **rA** là tỉ giá của đơn vị tiền tệ **A** theo đơn vị tiền tệ cơ sở);
* Chuyển đổi lượng tiền **MS** của đơn vị tiền tệ cơ sở thành lượng tiền **MB** của đơn vị tiền tệ **B (MB = MS \* rB**, với **rB** là tỉ giá của đơn vị tiền tệ **B** theo đơn vị tiền tệ cơ sở).

|  |  |
| --- | --- |
| VND | 22505 |
| JPY | 111.8 |
| EUR | 0.89 |
| GBP | 0.77 |

Thiết kế ứng xử chuyển đổi tiền tệ cho các đơn vị tiền tệ, sao cho ứng dụng có thực hiện chuyển đổi tiền giữa 2 loại đơn vị tiền tệ bất kỳ. (2.5đ)

Cài đặt và test ứng xử chuyển đổi các loại tiền tệ dưa vào bảng tỉ giá chuyển đổi theo đơn vị tiền tệ cơ sở là USD.

1. Nếu người ta muốn tạo ra một ứng dụng tổng quát có thể được sử dụng không chỉ để chuyển đổi tiền tệ, mà còn thực hiện chuyển đổi các đơn vị khác (khoảng cách: cm thành inch, Trọng lượng: kg thành pound, nhiệt độ: Celsius sang Fahrenheit, v.v.). Mẫu thiết kế nào phù hợp để đạt được sự linh hoạt này. Áp dụng vào thiết kế cho giải pháp bài toán, và cài đặt thêm chuyển đổi đơn vị khoảng cách: meter (mét) thành mile (dặm) 1 miles = 1609.344m, và Yard (thước Anh) 1 Yard = 0.9144m. (1đ)