Lab 14: Enumerations

In this lab, you will create an enum and a class that uses it.

Objectives:

- To write an enum

Part 1: Writing an Enum

Steps:

_1. Switch Eclipse to the Java perspective if not already in it.

_2. Create a new Java Project named **Lab14**.

_3. Right-click on the new project and choose *New* - *Enum* to start the wizard.

On the first wizard page, for the *Package*, enter **part1**.

For the *Name*, enter **Color**, the press Finish.

_4. Complete the enum by defining enumeration constants for red, green, blue, black and white.

_5. Create a class named **Automobile** in the *part1* package, then complete it:

a. Define fields for **weight** and **model**, using types of your choice.

b. Define a field of type Color (the enum) named **color**.

c. Generate get/set methods for the fields.

d. Generate (or write) a toString() method for the class.

_6. Create a new class named **Test** in the *part1* package with a *main()* method.

In main(), create a couple Automobile objects and initialze all of the their fields to values of your choice. Then call toString() on each object.

_7. Run and test the main program and ensure that the enum worked properly.

_8. Use *File* - *Close All* to close all open editor windows.

Part 2: Enhancing the Enum

Steps:

_1. Right-click on the *part1* package and choose *Copy*, then right-click on the *src* folder and choose *Paste*. Name the copied package **part2**.

_2. In the *part2* package, in the enum, add a private field of type *int* named **rgb**.

Generate a "get" method for the field.

_3. Write a private constructor in the enum that accepts an *int*. The constructor should assign the input parameter to the *rgb* field you created in the last step.

Copyright © Descriptor Systems 2016. All Rights Reserved. Materials may not be reproduced in any form without prior written consent of Joel Barnum.

Save the file - note that you now have errors on the enum constants - the enum no longer has a default constructor, so the constants must use the new constructor you just wrote.

_4. Modify each of the constants so they use the constructor. Pass the following values: 0x000000 for black, 0xffffff for white, 0xff0000 for red, 0x00ff00 for green and 0x0000ff for blue.

_5. In the Automobile class, modify the toString() method so it also prints out the automobile's color RGB value.

Note that you can use the Integer.toHexString() method to convert the RGB integer to a hexadecimal value.

_6. Run the main program and ensure that you see the RGB values.

Part 3: More Practice with Enums

In this part, you will create an enumerated type that represents a traffic light. Each enum constant will contain the duration (in seconds) of the specified light color. You will also create an enumerated type that represents a traffic light's condition (e.g. whether it's functional or not).

You will then write a testing program.

Steps:

_1. Create a new enum named **LightState** in the *part3* package.

Complete the enum according to this specification:

a. The states of a traffic light and their durations are *red*, which lasts for 120 seconds, **yellow**, which lasts for five seconds, and **green**, which lasts for 150 seconds.

b. The enum should define a constructor that accepts the duration.

c. The enum should define a "get" method that returns the duration for a state.

_2. Define another enumeration named **LightCondition** with constants for **functional**, **flashing** and **broken**.

_3. Create a new class named **TrafficLight** in the *part3* package. The class doesn't need a *main* method.

Complete the class according to the following specifications:

a. The traffic light should have fields for the light state and condition as well as field(s) that describe the intersection where the light is located.

b. Generate get/set methods for the fields.

c. The TrafficLight class should have a constructor that accepts parameters for all of the fields.

d. The TrafficLight class should have a toString() method.

_4. Create a new class named **TestTrafficLight** in the *part3* package. The class should have a *main* method.

Complete main() so it creates a TrafficLight object and displays it to the console. It should also print the duration of the light's current state.

Copyright © Descriptor Systems 2016. All Rights Reserved. Materials may not be reproduced in any form without prior written consent of Joel Barnum.