Assignment 1: Simple Java Application

Object-oriented programming Fall 2004 (4.0 / 5.0 points)

Uppsala University
Department of Information Technology
Division of Computing Science

Version 1.1, 10/11/2004
Assistants: Jesper Wilhelmsson and Ruslan Fomkin

1 Introduction

This assignment helps You to get to know Java: The Java language and the Java SDK. The assignment will require You to know Java syntax, to be able to write code using the Java language and comment the code in a standardized way. You will need to use commands from Java 2 SDK to compile the code (javac), to execute binaries (java), and to create a specification of the code (javadoc).

After the assignment You will be able to present a program structure in a diagram in the UML notation. You will use a UML diagram to present a design of your new application.

2 Preparations

Read about main object-oriented concepts, classes and relationships between them such as inheritance and associations. Read about overloading and overriding methods. Read about Java syntax and about Java SDK and its commands: javac, java, javadoc. Study UML notation.

Download the source code for this assignment from the course web page. Try to compile (command javac) and execute (command java) the given code in this assignment. Try to create a standard documentation of the code by using the command javadoc. Have a look at the created html-files.

3 Assignment description

The application is used to draw geometrical shapes. The given source code contain classes presenting circles and rectangles and classes that are used to draw the figures. You are encouraged to compile and execute the code and also to create a documentation and look at it before you start with the exercises.

3.1 Exercise 1: Extention

Your task is to extend the existing application by overloading the constructors of the classes Circle and Rectangle and by creating a subclass, Square, to the class Rectangle. You should also make use of the extentions by adding a few lines of code to the class GeoPaint.

1http://www.it.uu.se/edu/course/homepage/oop
1. The given constructors of the classes `Circle` and `Rectangle` create a new object in a given position with a random size. The new constructors that You should add to these classes will create objects with a given fixed size in a given position (Given = given by arguments).

2. A square is a kind of Rectangle it is therefore natural to specialize the class `Rectangle` with the class `Square`. Your task is to create a new class, `Square`, that extends the class `Rectangle`. In the new class You should create both constructors that create a square object in a fixed position with a random size and in a fixed position with a fixed size. To avoid code duplication, both constructors should use one of the constructors from the class `Rectangle`. All squares should be blue, this means you have to override the method `getColor` to give the square this property.

3. After You have extended the application it is good to test your new code and see how it works. The button `Square (Random)` should create new square with random size, the button `Square (fixed)` should create new square with size given by the slider, and the button `Circle` should create new circle with size given by the slider. To do so you need to add some code to the class `GeoPaint` so that it will use the new constructors and the new class. The changes should only involve the method `actionPerformed`. Note that there already are buttons in the application for the square objects, you only have to use them. More information is found in the comments in the code of the method `actionPerformed`.

3.2 Exercise 2: UML

In this exercise You need to present the extended application using UML notation. The diagram that You create on paper by hand or electronically using some diagram tool should contain the classes that builds this application, with their attributes and methods (including constructors), and all relationships between them. You should also present those external classes (without attributes and methods), e.g. the standard java classes, that the classes of this application have relations to and present relationships between the external classes and the application’s classes.

3.3 Exercise 3: Questions

- The class `Board` access the collection `things` belonging to the class `GeoPaint`. Why is this possible? Can other classes, for instance `Circle`, access `things`?
- What is wrong with this constructor:

  ```java
  public Square(Coordinate position)
  {
    int size = (int)(Math.random() * RANDOM_SIZE\_LIMIT) + MIN\_RANDOM\_SIZE;
    super(position, size, size);
  }
  ```

- The method `paint` implemented in the class `Rectangle` always invokes the method `getColor`. The class `Square` does not have its own implementation of the method `paint` so the method `paint` implemented in the class `Rectangle` is invoked when a square should be drawn. What instance of the method `getColor` is invoked by the method `paint` when a square is drawn? Why? Which instance of the method `getColor` is invoked when a rectangle is drawn?

4 Hand In

You should hand in your report to the mail box of corresponding assistant. The hard copy of the report should contain links to the source code and class files with solution of Exercise 1, the UML diagram for Exercise 2 and the answers to the questions of Exercise 3. Do not forget to state the name of the course, the assignment number (1) and names of both members of the group.