

QUIZ 2

Issue Date : 05.04.2024 - Friday

Due Date : 07.04.2024 - Sunday (23:00)

Advisor: Umut Güngör, R.A. Aslı TAŞGETİREN

Programing Language : Java 8



1 Introduction

Inheritance and polymorphism are fundamental concepts in Object-Oriented Programming (OOP) that facilitate code organization, reusability, flexibility, and modeling real-world hierarchies. Inheritance refers to the ability of a class (referred to as the subclass or derived class) to inherit properties and behaviors from another class (known as the superclass or base class). This allows the subclass to reuse code from the superclass, promoting code reusability and reducing redundancy. Polymorphism, which builds on inheritance, refers to the ability of objects to take on multiple forms or behaviors based on their context. This concept allows different classes to be treated as instances of a common superclass, providing flexibility and allowing for more generic code. Together, they create a powerful combination for flexible and reusable object-oriented programming.

2 Classroom Decoration

The construction of a new building in a university has been completed and you have been asked to calculate the cost of decorating this building. There are two types of classrooms in this building: circular and rectangular classrooms. You are going to decorate both wall and flooring of the classrooms. You can do this decoration with three types: paint, wallpaper, and tiles. However, there are some restrictions in this decoration. In circular classrooms, the decoration of walls can only be done with paint. There is no restriction on the decoration of walls in rectangular classrooms. The decoration of floors can only be done with tiles for both classrooms. You are going to calculate the cost of decoration of classrooms and the total cost. Cost of the decoration of wall will be calculated this way: If the wall will be decorated with tiles, firstly, you will find how many tiles there can be on the wall then, you multiply it with the cost of one tile which will be given in the input file. If the wall will be decorated with paint or wallpaper, you will calculate the area of the wall and multiply it by the cost per square meter of relevant type which will be given in the input file. Similarly, cost of the decoration of flooring will be calculated this way: You will find how many tiles there can be

on the flooring then, you multiply it with the cost of one tile. In your implementation, you must use Builder Design Pattern.

3 Builder Design Pattern

The builder design pattern simplifies constructing complex objects with numerous parameters. It separates object creation logic from the object itself. You use a dedicated "builder" class to configure the object step-by-step, specifying optional or mandatory parameters. Finally, you call a "build" method on the builder to create the final object. This approach enhances code readability, facilitates handling optional parameters gracefully, and can even promotes the creation of immutable objects for improved data integrity. For more information, you can look at this [link](#).

4 Input Output Format

You will have two input files, which are items.txt and decoration.txt, and one output file which is output.txt.

4.1 items.txt

This input file includes the classrooms in the building and different types of decorations. If a line starts with **CLASSROOM** it indicates a classroom and it includes classroom's name, shape, width, length and height. If a line starts with **DECORATION** it indicates a decoration. If the decoration type is **Tile**, it will include decoration's name, type, price per tile, and area of one tile. If the decoration type is either **Paint** or **Wallpaper**, it will include decoration's name, type, and price per square meter.

Format of items.txt:

CLASSROOM[tab]**name**[tab]**shape**[tab]**width**[tab]**length**[tab]**height**

DECORATION[tab]**name**[tab]**type**[tab]**price**

OR

DECORATION[tab]**name**[tab]**type**[tab]**price**[tab]**area**

Example lines form items.txt:

```
CLASSROOM    C1      Circle 5      5      10
CLASSROOM    C2      Rectangle 15    5      10
CLASSROOM    C3      Rectangle 5     15     10
CLASSROOM    C4      Circle 10     10     8
DECORATION    D1      Paint 10
DECORATION    D2      Tile 100      3
DECORATION    D3      Wallpaper 50
```

First line indicates that Classroom C1 is Circular. Its width is 5 m, length is 5 m and height is 10 m.

Fifth line indicates that type of Decoration D1 is Paint and its cost per square meter is 10 TL.

Sixth line indicates that type of Decoration D2 is Tile. Its cost per tile is 100 TL and area of a tile is 3 m².

4.2 decoration.txt

This input file includes which classroom will be painted with which decoration. It includes classroom name, decoration name for the walls and decoration name for flooring decoration.

Format of decoration.txt:

classroomName[tab]nameOfWallDecoration[tab]nameOfFlooringDecoration

Example lines form decoration.txt:

```
C1      D1      D2
C2      D2      D2
C3      D3      D2
C4      D1      D2
```

First line indicates that classroom C1's wall is decorated with decoration D1 and its flooring is decorated with decoration D2.

4.3 output.txt

The output file should include price of the each classroom's decoration and the total price. You should round decimal numbers to the integer values that are equal or greater than the actual values before writing to the output file. (You can use Math.ceil() function.)

Example lines form output.txt:

```
Classroom C1 used 158m2 of Paint for walls and used 7 Tiles for flooring, these costed 2271TL.
Classroom C2 used 134 Tiles for walls and used 25 Tiles for flooring, these costed 15900TL.
Classroom C3 used 400m2 of Wallpaper for walls and used 25 Tiles for flooring, these costed 22500TL.
Classroom C4 used 252m2 of Paint for walls and used 27 Tiles for flooring, these costed 5214TL.
Total price is: 45885TL.
```

In the output file, you should show Paint and Wallpaper by how much area they decorated. But, for Tile, you should show how many of them are used to decoration.

Execution and Test

While testing the input files' and output file's relative path (items.txt, decoration.txt and output.txt) should be given as arguments, *in that order*. You should do the following steps:

- Upload your java files to your server account (dev.cs.hacettepe.edu.tr)
- Compile your code (javac8 *.java, or javac8 Main.java)
- Run your program (java8 Main items.txt decoration.txt output.txt)
- Control your output file output.txt, it should have same data and format as the one given to you with your quiz.

Grading Policy

Task	Point
Correct output	100*
Total	100

***Even though it looks like only getting correct output will be enough to get 100% in this assignment, you are still required to use Inheritance, Polymorphism, Builder Design Pattern and other rules stated in Notes and Restrictions section. If you do not follow these rules you will face point deductions even if your output is 100% correct.**

Submit Format

File hierarchy must be zipped before submitted (Not .rar, only .zip files are supported by the system)

- b<studentid>.zip
 - <src>
 - Main.java, *.java

Notes and Restrictions

- **Since the main objective of this experiment is to gain experience on Object Oriented Programing, any solutions that does not use Inheritance and Polymorphism will not be accepted.**
- Do not miss the submission deadline.
- You must obey given submit hierarchy and get score (1 point) from the submit system, if not, as stated in BBM 104 Laboratory Rudiments you will lose 10% of your grade.
- Save all your work until the quiz is graded.
- Compile your code on DEV server before submitting your work to make sure it compiles without any problems on our server.

- Source code readability is a great of importance for us. Thus, write READABLE SOURCE CODE, comments and clear MAIN function. This expectation will be graded as “clean code”.
- Regardless of the length, use UNDERSTANDABLE names to your variables, classes and functions. The names of classes, attributes and methods should obey Java naming convention. This expectation will be graded as “coding standards”.
- You can ask your questions through course’s piazza group and you are supposed to be aware of everything discussed in the piazza group. General discussion of the problem is allowed, but DO NOT SHARE answers, algorithms, source codes and reports.
- All assignments must be original, individual work. Duplicate or very similar assignments are both going to be considered as cheating.