CSC111 Lab

Object Oriented Programming – Lab III

---- Lab 8 ---

Create a project Lab08

Q1) Modify class **Student** from lab 6. The class student had the following private attributes:

name: **String** containing first name of student.
age: **int** Student's age.
GPA: **double** Student's GPA

In addition to the method printInfo() we added two constructors and three setters and three getters:
Student()
Student(String name, int age, double GPA)
setName(String name):void
getName() :String
setAge(int age):void
getAge() :int
setGPA(double age):void
getGPA() :double
printInfo(): void

Student
-name: String -age: int -GPA: double
+ Student () + Student (String name, int age, double GPA) + setName(String name):void + getName() :String + setAge(int age):void + getAge() :int + setGPA(double age):void + getGPA() :double + printInfo(): void

Write a class **testStudent** that does the following:

- Declare two Student objects **s1** and **s2** from the class Student.
- Use empty constructor on s1 and read his information using setters.
- Print s1 info using method printInfo.
- Use second constructor on s2 to set initial values of name, age and GPA entered by the user.
- Then use the method printInfo() to print the information for s2.
- Print s1 info using getters.
- Find out who has the highest GPA and print his name.

Q2) Write a Java class that represent a building. The building has a number of appartments for rent, some appartments are normal and some are delux. The delux appartments are %20 more expensive than the normal ones. Here is the UML diagram:

Building
- apts: integer - normal: integer - delux: integer - rent: double
+ Building() + Building(int apts, double rent) + setApts(int apts):void + setRent(double rent): void + getApts():int + getRent():double + getNormal():int + getDelux():int + howManyRented():int + rentApt(int n, String type): boolean + printInfo():void

As shown in the UML diagram, write the class Building that has the attributes (all are private):

- **apts**: Total number of appartments in the building.
- **normal**: The number of rented normal appartments in the building.
- **delux**: The number of rented delux appartments in the building.
- **rent**: The monthly rent of a normal appartment.

The methods of this class are:

- **Building**() empty constructor
- **Buidling**(int apts, double rent) A constructor that sets **apts** and **rent** to the provided values. Sets **normal** and **delux** to zero.
- **setApts**(int **apts**): void. Sets the value of the attribute **apts** (since its private)
- **setRent**(double: rent): void Sets the value of the attribute **rent** (since its private)
- **getApts**():int. Returns the total number of appartments in the building
- **getRent**():double Returns the **rent** amount.
- **getNormal**():int. Returns the number of rented **normal** appartments.
- **getDelux**():int. Returns the number of rented **delux** appartments.
- **howManyRented**():int. Returns how many appartments are rented.
- **rentApt**(int **n**, String **type**):boolean. This method receives an integer **n** which is the number of appartments we like to rent and their **type**. Based on the value of type, you should increase the variable **normal** or **delux** accordingly. But first, you must check if you have n appartments available to rent. If you do, then rent the appartments and reurn true, else, return false.
- **printInfo**():void. print all information about the building in an organized manner as shown in sample run.

After writing class building, write a java class **testBuilding** to test your building class. In this class you should:

1. Declare an object **b** of class Building.
2. Read two building attributes: **apts** and **rent** from the user. No need to read normal and delux appartments numbers because they start as zero.
3. Ask the user how many **normal** appartments he would like to rent. If there is availability rent them and display a message stating that **n** appartments were rented. If not you should display the message "**Requested number of appartments exceeds availability**"
4. Ask the user how many **delux** appartments he would like to rent. If there is availability rent them and display a message stating that n appartments were rented. If not you should display the message "**Requested number of appartments exceeds availability**"
5. Print the building Information.
6. Feel free to try creating additional building objects, read their data and manipulate them.
7. exit.

SAMPLE RUN

```
Please enter Number of Appartments and rent amount :50 2000
How many normal appartments would you like to rent? 15
15 normal Appartments have been rented
How many delux appartments would you like to rent? 20
20 delux Appartments have been rented

================== Building Info ==========================
The Building has 50 appartment.
Only 35 have been rented.
15 normal appartments with rent = 2000.0 SR per month.
And 20 delux appartments with rent 2400.0 SR per month
===========================================================
```

SAMPLE RUN2

```
Enter number of appartments and rent amount :100 10000
How many normal appartments would you like to rent? 70
70 normal Appartments have been rented
How many delux appartments would you like to rent? 40
Requested number of appartments exceeds availability

================== Building Info ==========================
The Building has 100 appartment.
Only 70 have been rented.
70 normal appartments with rent = 10000.0 SR per month.
And 0 delux appartments with rent 12000.0 SR per month
===========================================================
```

Q3) Design a class named **Stock** for a company stock system.

The class contains:

- Private data fields **symbol**, **name**, **currentPrice** and **previousClosingPrice**.
- A method named **getChangePercent**() that returns a percentage of how much the price of the stock has changed.
- Methods **setName**(), **setSymbol**(), **setCurrentPrice**() and **setPreviousClosingPrice** that set the new values to the variables.
- Methods **getName(), getSymbol(), PreviousClosingPrice**() and **getCurrentPrice**()that return the values of the variables.

Draw the UML diagram for the class and then implement the class. Write a class **testStock** that prompts the user to enter the variables **symbol**, **name**, **currentPrice** and **previousClosingPrice** and displays stock symbol, company name, the current price and the percentage in which the price has been cahnged by.

Sample Run 1

```
Enter symbol of stock: APPL
Enter company name: Apple
Enter previous closing price: 100
Enter curret price: 108
For the stock APPL of the company Apple :
Previous Closing Price: 100.0
Current Price: 108.0
Price Change: 8.0%
```

Solution

Q1)

```
public class Student {

	private String name;
	private int age;
	private double

	// Methods

	public Student() {
	}
	public Student(String name, int age, double GPA) {
		this.name = name;
		this.age = age;
		this.GPA = GPA;
	}

	public void setName(String name){
		this.name = name;
	}
		public String getName(){
		return name;
	}
		public void setAge(int age){
		this.age = age;
	}
		public int getAge(){
		return age;
	}
		public void setGPA(double GPA){
		this.GPA = GPA;
	}
		public double getGPA(){
		return GPA;
	}

	public void printInfo() {
		System.out.println("------------------------");
		System.out.println("Student name: "+name);
		System.out.println("Student age: "+age);
		System.out.println("Student GPA: "+GPA);
		System.out.println("------------------------");
	}
}
```

```
======================  Class testStudent  ==========================

import java.util.Scanner;
public class testStudent {
	public static void main(String[] args) {
		Scanner kb = new Scanner (System.in);

		// Using empty constructor
		Student s1 = new Student ();
		System.out.print("Please enter the name, age and GPA ");
		s1.setName(kb.next());
		s1.setAge(kb.nextInt());
		s1.setGPA(kb.nextDouble());

		// Print student info
		s1.printInfo();

		// Using second constructor
		System.out.print("Please enter the name, age and GPA ");
		Student s2 = new Student( kb.next(), kb.nextInt(), kb.nextDouble() );

		// Print student info
		s2.printInfo();

		// Print student info using getters
		System.out.println("The first student name is "+s1.getName()+" and his
age is "+s1.getAge()+" his GPA is "+s1.getGPA());

		// Which student has a higher GPA?

		if (s1.getGPA() > s2.getGPA())
			System.out.println(s1.getName()+" has a higher GPA than
"+s2.getName());
		else
			if (s1.getGPA() < s2.getGPA())
				System.out.println(s2.getName()+" has a higher GPA than
"+s1.getName());
			else System.out.println(s1.getName()+" and "+s2.getName()+" have
same GPA");

		kb.close();
	}
}
```