

Homework 8

Defining New Classes

Do this homework for attendance
at the eighth laboratory session

Exercise 1

Define a class, called *Circle*, having the methods *getArea()*, which returns the area of a circle and *getCircumference()*, which returns the circumference of a circle. The constructor has as arguments the coordinates of the center (two integers) and the radius of the circle (a double).

Exercise 2

Define a class called *Student*. Each student has a name and a total number of points obtained at the exams. Define an appropriate constructor and four methods: *getName()* which returns the name of a student; *addExam(int mark)* which adds the mark for each exam; *getTotal()* which obtains the total number of points; *getMeanMark()*, which returns the mean of all marks.

Exercise 3

Define a class called *Product*. Each product has a name and a price (in ROL). Write the constructor and the following methods: *displayProduct()*, *getPrice()* and *setPrice()*. Add two other methods: *getPriceInRON()* and *setPriceInRON()* which handles RONs. Write a program that creates two products and displays them (in ROLs), then raises their prices with 10.25 RON, displays them again (in RONs) and finally only their prices in RONs.

Exercise 4

Write a program that verifies if the date introduced by the user is correct. The format of the date is dd/mm/yyyy, and the year must be between 1901 and 2099. Use JOptionPane for interacting with the user.

Write a variant of the program when the input data (day, month, and year) are read from the command line and the result is displayed on the screen.

Exercise 5

Define a class, called *Rational*, which represents rational numbers with their arithmetic operations. Use integer variables to represent the private instance variables of the class: numerator and denominator. Write a default constructor which initializes the numerator with 0 and the denominator with 1. Another constructor with two integer parameters will keep the fraction in its reduced form (for instance, the constructor with 2 and 4 as parameters will be stored into the object as 1/2).

The class *Rational* will contain public methods which will implement the following operations on rational numbers:

add()- addition of two rational numbers; the result will be a rational number stored in the reduced form

sub()- subtraction of two rational numbers; the result will be a rational number stored in the reduced form

mult()- multiplication of two rational numbers; the result will be a rational number stored in the reduced form

div()- division of two rational numbers; the result will be a rational number stored in the reduced form

intFormat()- displaying of the rational number in the numerator/denominator format

realFormat()- displaying a rational number in the floating point format

Define another class, called *TestRational* for testing the *Rational* class.

Exercise 6

Define a class *Complex* which represents complex numbers with their arithmetic operations. Use double variables to represent the private instance variables of the class: real part and integer part.

The complex numbers have the form $realPart + i * imaginaryPart$, where $i^2 = -1$.

Define two constructors, the default one that initializes at 0.0 the instance variables and the other that initializes them with the values of parameters.

Write public methods for the following operations:

add() -addition of two complex numbers; the result is a complex number, too

sub() -subtraction of two complex numbers; the result is a complex number, too

mult() - multiplication of two complex numbers; the result is a complex number, too

div() - division of two complex numbers; the result is a complex number, too

toString() - displaying a complex number in the format (a,b) where a is the real part and b is the imaginary part of the complex number

Define another class, called *TestComplex*, for testing the *Complex* class.

Exercise 7

Define a package *numbers* containing the classes *Rational* and *Complex* defined in the exercises 5 and 6.

Write a class *TestNumbers* in the anonymous package with a *main()* method which tests the classes in the package *numbers*.