Programming Exercises - PRO1 - Session 04

Exercise 4.01

For each of the three classes (Person, MyDate and Payroll) that you created in the exercises from our last session (Exercise 3.01, 3.02 and 3.03) do the following:

a) Create a constructor with the necessary parameters to initialize the instance fields in the classes.
b) Create a no-argument constructor to initialize the instance fields to default values of your choice.
c) Create a toString() method, returning a meaningful representation of objects from the class.
d) Add the constructors and new methods to your UML class diagrams
e) Add the necessary changes to your existing test classes, to test the new functionality.

Exercise 4.02

Create a UML class diagram for a class Book. Then implement it in Java and test it. The Book class should have the following fields: author (String), title (String), price (double), and pages (int).

a) Create a 4-argument constructor initializing all fields.
b) Create get methods for all fields.
c) Create a set method for price.
d) Write a test program that creates two Book objects based on user input (use a Scanner object to input author, title, price, and pages), then lowers the price on both books by 20%, and finally prints out the information about the two books.

Exercise 4.03 [Gaddis] Programming Challenge 2, p. 210

Exercise 4.04 [Gaddis] Programming Challenge 4, p. 210

Exercise 4.05 [Gaddis] Programming Challenge 8, p. 212

Exercise 4.06

Create a class Student that has:

a) Three instance variables: a name (String), a gender (char - then when you create objects in the main method, use the values 'M' or 'F', and perhaps any other letters you identify as), and a student number (int).
b) A 3-argument constructor setting all three instance variables.
c) A 2-argument constructor with a name and a gender as argument. Assume that the student number is 0 if not set.
d) Set methods for name and student number. Call the methods setName and setStudentNumber.
e) Get methods for all three instance variables. Call the methods getName, getGender and getStudentNumber.

Then create a test class (StudentTest) with a main method and test the class Student:

f) Create at least three Student-objects.
g) Call all the methods you made in class Student, i.e. both constructors, all set and get methods.
h) Print out all information of each Student-object.

Exercise 4.07

Modify class Student from the exercise above and:

a) Add a method toString() that return a String with all information of a Student-object. As an example calling the method toString() on a Student-object with name = "Bob", gender = 'M' and student number = 2342 could return the following string: "Bob, 'M', 2342".
b) Change the test class StudentTest so that you call method toString in the print-statements where you print out all information of each of the students.