Assignment 8, year 2011

Create the *Statistics* class with the following attribute:

- *percentagePassed* representing the percentage of students with a passing exam mark (e.g., 45.3)

Create the *MarksArray* class with the following elements:

- An attribute representing an array of students from the same class; each student has one mark, and the marks are represented with the following letters: 'A' for excellent, 'B' for very good, 'C' for good, 'D' for satisfactory, and 'N' for unsatisfactory.
- An attribute representing the number of marks (students) in the class (array); initial value is zero.
- The *classStatistics* attribute, as object of the *Statistics* class; the attribute should be initialized.
- A constructor receiving, as its input parameter, the maximum number of marks (students), and initializing the array to this capacity. If the input parameter is less than or equal to zero, an error message should be printed, and the array should be initialized to the maximum capacity of 25 marks (students).
- A method for adding a new mark to the array. The new mark is the input parameter of this method, and it should be added to the array only if the array is not already full, and if the mark has one of the following values: 'A', 'B', 'C', 'D', or 'N'. If any of these two conditions does not hold, an error message should be printed to the screen.
- A method that for each mark, prints to the screen the percentage of students with that mark. The method should also compute the percentage of students with passing marks (i.e., percentage of marks different than 'N') and assign the computed value to the *percentagePassed* attribute of the *classStatistics* object.
- A static method that receives, as its input parameter, an array of characters representing marks of a few classes. This array is completely full (i.e., filled up to its capacity). Each class has exactly 25 students; so, first 25 elements of the input array are marks of the students of the first class, the next 25 elements are marks of the students from the 2nd class and so on. The method should check if there is a class where all students fell the exam (i.e., all 25 students got the 'N' mark). If there is at least one such class, the method returns true, otherwise it returns false.

Create the *TextMarksArray* class that creates one object of the *MarksArray* class with maximum capacity of 10 students (marks), and adds to it the following marks: 'A', 'B', 'C', 'N', 'N', and 'D'. It should also print to the screen for each mark, the number of students with that mark.

Assignment 5, year 2012

Create the *Airplane* class with the following elements:

- An array attribute representing the seats of the economy class; a seat is free if the corresponding element of the array has value TRUE, and occupied if the value of the corresponding array element is FALSE.

- An array attribute representing the seats of the business class; a seat is free if the corresponding element of the array has value TRUE, and occupied if the value of the corresponding array element is FALSE.

- A constructor receiving as its input parameters the number of seats in the economy class and the number of seats in the business class; it uses these values to initialize the two arrays to the corresponding capacities. If *any* of the two parameters is less than zero, the array of economic class seats should be initialized with the capacity of 120, and the array of business class seats with the capacity of 60; in addition, an error message should be printed on the screen. In any case, all seats in the airplane should be set to free.

- A method for seating a passenger. The method receives, as its input parameter, a boolean value with value TRUE if the passenger is to be seated in the business class, and FALSE if he/she is to be seated in the economy class; the other input parameter is the seat number for the passenger (seat numbers start from zero). Based on the values of the input parameters, the method seats the passenger to the appropriate seat. If that seat is occupied, the method prints an error message.

- A method that computes and prints to the screen a report about the flight. In particular, it computes and prints the following: the percentage of occupied seats in the business class, the percentage of occupied seats in the economy class, and the income from the sold tickets. The price of a business class seat is 500 euros, while the price for an economy class seat is 250 euros. The income is computed only for the occupied seats.

- A method that checks if in the economy class there is a row with 3 consecutive seats that are free; if there is such a row, the method returns TRUE, otherwise it returns FALSE. All rows are of the same length; in particular, each row has precisely 6 seats, so that first 6 elements (seats) of the array are in the 1st row, next 6 seats are in the 2nd row, and so on.

Create the *TestAirplane* class that creates 3 objects of the *Airplane* class. The first airplane has 120 seats in the economy and 60 seats in the business class, the second has 50 seats in each class, and the 3rd one only has 30 seats in the business class. Seat passengers in the 3rd plane, on the business class seats with numbers 1, 5 and 15.