American University of Armenia
CS 120 Intro to OOP
Spring 2019

Homework Assignment 3

1. **(10 points)** Write a Java program that inputs the radius of a circle and outputs its circumference and its area. Your program should consist of two files:

   - `Circle.java`, containing the definition of the `Circle` class with corresponding instance variables and methods;
   - `CircleDemo.java`, containing the demo class with the `main` method that illustrates the use of the `Circle` class.

   If both files are put in the same directory (folder) compiling and running `CircleDemo.java` in the usual way, i.e. `javac CircleDemo.java; java CircleDemo` should work.

2. **(15 points)** Write a Java program that approximates the percentage of prime numbers in some range $1..N$. First of all, your program should include a method that checks if a given number is prime. Your program should follow the procedure below:

   (a) Read in the integer N

   (b) For K equalling each of 100, 200, . . . , 900, 1000, do

   (c) Generate K random numbers in the range $1..N$, check how many of them are prime and output the corresponding percentage.

   What is the advantage of this approach over checking every single number in the range $1..N$ and finding the accurate percentage? How does the accuracy of the calculation change when we move from $K = 100$ to $K = 1000$?

3. **(15 points)** Write two variants (iterative and recursive) of a Java program that reads a natural number n $(1 \leq n \leq 80)$ and prints the n-th Fibonacci number.

4. **(10 points)** Imagine the following game: you start with a positive integer number a. Then taking the sum of the digits of a to get the number b. Then taking the sum of the digits of b you get the number c. Continuing in this manner, you eventually get a single-digit number (or do you?) and stop the game there.

   Write a Java program that reads a positive integer number a and simulates this game by printing the resulting numbers at each step. Once a single-digit number is reached, the program stops and prints the number of steps taken. Does your program always terminate? Explain your answer.

5. **(10 points)** Write a Java program that inputs the daily temperatures for one month (30 days) and outputs the following information:

   - the hottest and the coldest days of the month (both days and the corresponding temperatures);
   - the average temperature of the month;
   - the temperature difference between the hottest and coldest days of the month.

   Similar to Task 1, your program should consist of two source files: one with a corresponding class definition and one with the demo program.

6. **(20 points)** Write a Java program that reads the coordinates of two 3D vectors and prints the coordinates of their vector product. Your program should include a class `Vector3D` definition for representing 3D vectors and a method that finds the vector product of two 3D vectors, i.e. it takes two parameters of type `Vector3D` and returns a `Vector3D` object.

7. **(20 points)** Write two variants (iterative and recursive) of a Java program that reads two strings a and b ($1 \leq |b| \leq |a| \leq 100$), and prints the number of substrings of a that are cyclic shifts of b. The cyclic shifts of a string $b = c_1c_2 \ldots c_n$ are $c_1c_2 \ldots c_n$; $c_2c_3 \ldots c_nc_1$; $c_3 \ldots c_nc_1c_2$; and so on until $c_nc_1c_2 \ldots c_{n-1}$. The substrings of a string $a = d_1d_2 \ldots d_m$ are any $d_id_{i+1} \ldots d_{j-1}d_j$ where $1 \leq i \leq j \leq m$.

   For example, on input `abcabc abc` your program should output 4, while on input `abcabc acb` it should output 0.