OBJ2000 Exam – December 2020

Task 1 (10 points).

Write a Java program to calculate the percentage of the correctly answered questions on a student's test, given the total number of questions on the test and the number of questions the student got right.

The total number of questions on the test and the number of questions the student got right should be given to the user to input and later stored in the variables `totalQuestions` and `numRight`, respectively. Also, note that the percentage should include the fractional part.

E.g. 1: if the value of `numRight` is 4 and the value of `totalQuestions` is 10, the code prints 40.0%.
E.g. 2: if the value of `numRight` is 9 and the value of `totalQuestions` is 15, the code prints 60.0%.
E.g. 3: if the value of `numRight` is 6 and the value of `totalQuestions` is 9, the code prints 66.7%.

Task 2 (10 points)

Write a Java program that calculates the following sum for the k entered by user.

$$S = 1^2 + 3^2 + 5^2 + \cdots + (2K + 1)^2$$

Task 3 (15 points)

Given two integer variables `a` and `b`, write a method

```
public int firstMultiple(int a, int b)
```

that returns the first multiple of `a` that is greater than `b`. Assume that the initial value of the variables `a` and `b` are positive integers.

E.g. 1: if the value of `a` is 2 and the value of `b` is 8, the method returns 10.
E.g. 2: if the value of `a` is 3 and the value of `b` is 5, the method returns 6.
E.g. 3: if the value of `a` is 6 and the value of `b` is 4, the method returns 6.

Task 4 (15 points)

The greatest common devisor (gcd) of numbers is the largest positive integer that divides each of the integers. Write a Java method that returns value of gcd.

```
public int gcd(int a, int b, int c)
```

E.g. 1: for a = 8, b = 12, c = 16 the method should return: 4

E.g. 2: for a = 11, b = 17, c = 10 the method should return: 1

E.g. 3: for a = 54, b = 26, c = 12 the method should return: 6

E.g. 4: for a = 5, b = 10, c = 20 the method should return: 5

Task 5 (15 points)

Write a method

```
public String specialConcatenate(String str1, String str2)
```

that will concatenate the two strings without including the first character of each of them. The method should return the resulting string. Assume that the initial value of variables `str1` and `str2` are with at least one character.

E.g. 1: if the value of `str1` is "Hello" and the value of `str2` is "There", the code prints ellohere.
E.g. 2: if the value of `str1` is "java" and the value of `str2` is "code", the code prints avaode.
E.g. 3: if the value of `str1` is "shotl" and the value of `str2` is "java", the code prints hotlava.

Task 6 (35 points).

The task is to create a Java program that supports the classification of documents kept in FBI (Federal Bureau of Investigation) archive.

1. Create a class named *Docs* that contains data fields for the *archive number*, *title*, *subtitle*, and *year*. Include *Get* and *Set* methods for these fields (2 points).
2. Class *Docs* should have two method constructors that assign values to the specific fields. One constructor with two arguments (*number* and *title*) and one with three arguments (*number*, *title*, *subtitle*) (3 points). *Year* should be updated in the constructors with the current year (1 point).
3. Class Docs should offer *print* method that writes a message about the title and subtitle of the document (3 points).
4. Create two subclasses that will extend class *Docs* and which will be named: *Reports* and *Contracts* (2 points).
5. *Reports* subclass should contain additional fields (1 point):
    a. One that holds a number of the *FBI department* that issued the report.
    b. One field that indicates whether the report is *confidential* or not.
6. *Reports* should have additional constructor that updates those two additional fields (3 points) in addition to the constructor from the superclass.
7. Add methods *getDepartment* and *isConfidental* that read and return values of the fields *department* and *isConfidental*, respectively (3 points).
8. Override the print method from the superclass to create method in *Reports* class to print information from all available fields (3 points).
9. *Contract* subclass should contain additional field that stores the name of the *partner* (that contract is signed with) (2 points).
10. Add constructor in *Contract* class updates the value of *partner* field (in addition to the constructor from the superclass) (2 points).
11. Use polymorphism to create two methods in Contract class to print information about the contract. One method should print only department, title and year. Other method should print all available information. These methods should not override the print method from the superclass (4 points).
12. Create simple *Test* class with main method, that will create instances of one Report and one Contract and print details from both (6 points).

Pay attention on the use of access modifiers and readability of the code. During the assessment of this task, special attention will be focused on the ability to implement basic principles of object-oriented programming: inheritance, encapsulation, and polymorphism.