Mock Test

This is only a mock test. You can test on how to create and name test programs correctly. The mock test is made up of six tasks. The first two tasks are already completed. Open these files and see how the tasks have been solved.

To check the correctness of all completed tasks, run the check.py program. The results will be saved to a text file.

**Please take into account that during the real test, you have to upload all created files/programs to the Moodle platform for assessment. You will have approximately 60-70 minutes to upload the files, depending on the number of tasks.**

**The tasks will be assessed automatically. Make sure that the names of the created classes, attributes and methods are consistent with the content of the task.**

(C1.java) Define a class with two attributes describing a person: name (String) and age (int). Apply data encapsulation. Define a constructor with the parameters name and adult to assign initial values of object’s attributes. Define get and set methods for each attribute. Then, define a method isAdult() that returns true if a person is an adult (person is at least 18 years old) or false otherwise. Finally, define a method that returns a string representation of the object (name and age, separated by comma). Sample result:

C1 o = new C1("Anna",21)  
o.getAge() à 21  
o.isAdult() à true  
o.setAge(17)  
o.isAdult() à false  
o.toString() à "Anna,17"

(C2.java) Define a class that allows you to create a counter of integer type. The initial value of the counter is 0. The class includes an increase() method that increases the value of the counter by 1 and a decrease() method that decreases the value of the counter by 1. Also, create two overloaded methods: increase(int n) and decrease(int n) that allow you to increase or decrease the value of the counter by n. Add a value() method that returns a counter value. Sample result:

C2 o = new C2()  
o.value() à 0  
o.increase()  
o.increase()  
o.decrease()  
o.increase(5)  
o.decrease(2)  
o.value() à 4

(C3.java) Define a class that contains two static methods: (1) isAlphabet(String t) that returns true if the letters in the text t are in alphabetical order or false otherwise and (2) hideText(String t) that returns the text t in which all characters except the first and last characters are converted to an asterisk. Sample result:

C3.isAlphabet("abegsw") à true  
C3.isAlphabet("abcmhsw") à false  
C3.hideText("apple") à "a\*\*\*e"  
C3.hideText("ok") à "ok"

(C4.java) Define a class with two static methods: even(int[] array) that returns the number of even values in the array, and the positiveOdd(int[] array) that returns the number of positive odd numbers in the array. Sample result:

int[] arr1 = {2,-6,5,8}  
int[] arr2 = {3,2,-5,4,1,-7}   
C4.even(arr1) à 3  
C4.positiveOdd(arr2) à 2

(C5.java) Define a class that contains two attributes: x and y, of integer type (int), describing the coordinates of a point on the plane. The class constructor contains two parameters and allows you to initialize the object's attributes. Create an isX() method in the class that returns true if the point is on the x-axis and false otherwise. Create an isY() method in the class that returns true if the point is on the y-axis or false otherwise. Add a method in the class to represent the object as text that returns the coordinates of the point in the format "P(x,y)". Do not add some space characters in this string. Sample result:

C5 o = new C5(3,0)  
o.isX() à true  
o.isY() à false   
o.toString() à "P(3,0)"

(C6.java) Define a class with a “title”attribute of a string type and a “pages” attribute of an integer type. Apply data encapsulation. Create set and get methods for each attribute. In the method that modifies the number of book pages, change the value of the object attribute only when the specified number of book pages is positive. Sample result:

C6 o = new C6()  
o.setPages(3)  
o.setPages(-4)  
o.getPages() à 3