Mock3 Test

Complete the following tasks. You have 80 minutes to complete the test. Upload the highlighted files in bold to the Moodle platform. Check that the names of the created files and created classes/methods/attributes match the contents of the task.

1. The **Person** class contains two attributes: name and surname. The initial values of the attributes are passed through the constructor parameters. Specify the text representation of the object, consisting of two uppercase letters of the first name and last name. Example:  
   Person("anna","may") 🡪 "AM"
2. The **Isogram** class includes the static isogram(String) method. The method returns true if the text parameter is an isogram or false otherwise. An isogram is a string where all characters are different. Example:  
   isogram("red sun") 🡪 true  
   isogram("blue water") 🡪 false  
   isogram("BLUE water") 🡪 true  
   isogram("my blue water") 🡪 false
3. The **Logic** class contains an attribute in the form of an array of logical values. The attribute is initialized through a constructor parameter. The opposite() method returns the number of array elements for which both adjacent elements have the opposite value. Example:  
   Logic([True,False,False]).opposite() 🡪 0  
   Logic([True,False,True,False]).opposite() 🡪 2  
   Logic([True,False,True,True,False,True,False,True,False]).opposite() 🡪 5
4. The **Number** class contains two attributes: system (int) and value (String). The system attribute specifies the positional system (from binary to decimal) for the value. Attribute values can be changed using modifying methods. The get10() method returns value in the decimal system as an integer or returns -1 if value does not match the given positional system. Example:  
   get10() 🡪 26 // system=5, value="101"  
   get10() 🡪 -1 // system=8, value="10283"
5. The **Cities** class includes an attribute in the form of an array with city names. The initial value of the attribute is passed through a constructor parameter. The filter(char) method returns an object of the Cities class with those cities whose names start with the given character. The cities() method returns a string consisting of the city names contained in the object attribute. Example:  
   Cities(["Warszawa","Sopot","Kielce","Szczecin"]).filter('S').cities() 🡪 "SopotSzczecin"
6. Based on the UML class diagram, define the **Pizza** class. In the Food class, setPrice is an abstract method. In this method, set the price of the pizza based on its size, according to the rule: size of the pizza - 10. The discount method returns the discount for the purchased pizza, according to the rule: when the price of the pizza is at least 30, the discount is 5%. The delivery method returns the cost of pizza delivery, which is 6, unless the size of the pizza is 50 or more, then the cost of delivery is 8. The cost of delivery may also include a tip for the supplier. The tip value can be passed via a method parameter.

Obraz zawierający stół

Opis wygenerowany automatycznie

1. The **Numbers** class includes an attribute in the form of an array of integers. The initial value of the attribute is passed through a constructor parameter. The class implements the interface:  
   interface Ok { boolean ok() }  
   The ok() method returns true when the even-indexed array elements contain even values and the odd-indexed array elements contain odd values. Otherwise, the method returns false. Example:  
   Numbers([6,7,6,1,4]).ok() => true  
   Numbers([2,5,2,8,4]).ok() => false
2. The **Product** class contains two attributes: name (String) and price (float). Implement a Comparable interface that will allow you to sort Product class objects by product name.