Wrocław, 19.11.2021

Structured and Object Oriented programming
Laboratory

Assignment #7: The First Class: exceptions

## Task 1

Implement a class with the name ProductTypeEx. It should contain the following data members:

- Name : a nonempty string of characters, null value is not allowed too
- itemPrice: double value with the price of a single item of that product type, must be >=0 and <=200
- comment
- priceChangeNo: int with the number of successful price modifications

All data members should be private.

The class should have at least the following methods:

- public Product(String  name, double price, String comment)
- public String toString()
- public String getName()
- public double getPrice()
- public boolean modifyName(String new_name) // ignore not valid names and return false
- public  boolean setPrice(double price) // the validity check should be the same as for the constructor
- public static ProductTypeEx[] createTestDataEx()
- public static ProductType[] createTestData()
- public static void testMe()

Remarks:

1. If the value of a constructor argument is outside the required range then throw an Exception object. The argument of its constructor must specify the type and the value that is not acceptable.
2. The method testMe attempts to create 5 objects of the ProductTypeEx class. Use both proper and not proper values of the constructor's arguments. Catch all exceptions caused by improper values of constructor arguments and display a useful message for the user.
3. The method createTestDataEx should return an array with 3 objects that is returned as its value.
4. The method createTestData should return an array with 5 objects that is returned as its value. Both proper and not proper objects are to be created.

Test thoroughly all of the above methods.

## Task 2

Implement a class ProductBox.
It should contain the following data members and methods:

```
private ProductType [] regularStore;
private ProductTypeEx[] exSore;

public ProductBox() {
}

public boolean setRegular(ProductType data) {
        regularStore=data;
}
public boolean setEx(ProductTypeEx data) {
        exStore=data;
}
// calculate the average value of objects from the regularStore array
// return Double.NaN whenever the average value does not exist
public Double calcAveragePrice()

// calculate the average value of objects from the exStore array
// return Double.NaN whenever the average value does not exist
public Double calcAveragePriceEx()
```

Do not forget to test thoroughly all of the above methods.

Andrzej Siemiński