
სემინარი 14

1. რა მოხდება testOperations() - ის გამოძახებისას?¹

```
/* Method: testOperations() */
/* Method: testOperations() */
/**
 * Passes two Objects of different classes (String and GOval) in separate
 * methods to test their behaviour afterwards.
 */

private void testOperations() {
    String s = "abc";
    GOval o = new GOval(1, 1);

    println(s);
    changeString(s);
    println(s);

    println("Width: " + o.getWidth() + " Height: " + o.getHeight());
    changeOval(o);
    println("Width: " + o.getWidth() + " Height: " + o.getHeight());
}

/* Method: changeOval(GOval o) */
/** Changes the size of passed GOval to 2, 2. */
private void changeOval(GOval o) {
    o.setSize(2, 2);
}

/* Method: changeString(String s) */
/** Changes the value of the passed string to "aaa" */
private void changeString(String s) {
    s = "aaa";
}
```

¹ პასუხი იხილეთ ბოლო გვერდზე.

2. შექმენით Mutable სტრინგის კლასი, რომელსაც ექნება შემდეგი მეთოდები:

```
public void setValue(String value); - შეცვლის არსებული სტრინგის მნიშვნელობას.  
public String getValue(); - დააბრუნებს არსებული სტრინგის მნიშვნელობას.  
public void replaceAll(String toReplace, String replaceWith); - ჩაანაცვლებს სტრინგში არსებულ ყველა toReplace მონაკვეთს replaceWith-ით.  
public void deleteSub(int start, int end); - წაშლის ყველა სიმბოლოს start-იდან end-მდე.  
public void append(String b); - ამ ობიექტს „მიანებებს“ b სტრინგს  
public int length(); - დააბრუნებს სტრინგის სიგრძეს.
```

კარგი იქნება, თუ თქვენს კლასს ექნება toString() მეთოდიც, რომელიც დააბრუნებს ზუსტად იგივეს, რასაც getValue() - ამ შემთხვევაში თქვენ მარტივად შეძლებთ ამ კლასის ობიექტის დაბეჭდვას და ჩვეულებრივ სტრინგზე კონკატენაციასაც!

პირველი ამოცანის ამოხსნა:

სტრინგის changeString-ში გატარებისას არაფერი შეიცვლება, რადგან სტრინგის კლასის ყველა ობიექტი immutable² არის. თან ჩვენ ამ მეთოდში მას მხოლოდ მისამართს ვუცვლით და ჩვენ რომელიც გადავცით, იმ მისამართზე არსებულ ობიექტში, არანაირი ცვლილება არ მოხდება და კონსოლი იგივე სტრინგს დაბეჭდავს.

GOval ობიექტის changeOval-ში გატარებით ყველა ცვლილება ობიექტის შიგნით ხორციელდება, ამიტომ ამ მეთოდის დასრულებისას ყველანაირი ცვლილება შეინახება ობიექტშივე და კონსოლი უკვე შეცვლილ პარამეტრებს დაბეჭდავს.

² Immutable (უცვლელი) არის ობიექტი, რომელშიც შექმნის შემდეგ ვერანაირ ცვლილებას ვერ შეიტანთ. String, პრიმიტიული ტიპები (boolean, char, int, double და ა.შ.).