

# Tēma 4

## Uzdevums 1 - Bool izteiksmes

### Soļi

- Izveidot klasi BoolExpressions.
  - Projekta navigātorā Right Click -> new -> class, nosaukums "BoolExpressions"
  - Atveras redaktors kurā tiek labota izveidotā klase.
- izveidot metodi test1() kura atgriež bool izteiksmes vērtību, arguments ir vesels skaitlis. Metode atgriež bool vērtību : arguments i satur vai nu 10 vai 20 vai 75.
  - piemērs. ievietojam šādu metodi:

```
boolean test1(int arg){  
    return (i==10 || i==20 || i==75);  
}
```

- izveidot metodi test2 kura atgriež bool vērtību, arguments ir vesels skaitlis. Metode atgriež bool vērtību : arguments i nesatur nedz 10 nedz 20 nedz 30
  - soļi analogiski, izveidojam jaunu metodi test2, parametri, atgriežamais tips pēc nosacījumiem.

- izveidot metodi test3() kura atgriež String vērtību, arguments ir vesels skaitlis. Metode atgriež String vērtību "Even" ja i dalās ar 2, pretējā gadījumā "Odd". Izmantojiet % operātoru.

- soli analogiski, izveidojam jaunu metodi test3, parametri, atgriežamais tips pēc nosacījumiem.

- lai noskaidrotu vai skaitlis dalās ar 2 bez atlikuma izmantojam % operātoru.

```
if(i % 2 == 0) // mainīgais i dalās ar 2 bez atlikuma.
```

- izveidot metodi test4() kura atgriež String vērtību, arguments ir vesels skaitlis. Metode pārbauda personas vecuma piederību vecuma intervālam:

"children" 0-10 g, "teenager" 11 - 20 g, "young" 21 - 30, "adult" 31-55, "senior" virs 55

- soli analogiski, izveidojam jaunu metodi test4, parametri, atgriežamais tips pēc nosacījumiem.

- izmantojam if-then-else operātoru lai noskaidrotu argumenta vērtības piederību vecuma grupai.

- metode atgriež attasto vecuma grupas teksta vērtību.

- izveidot metodi test5() kura atgriež bool vērtību, 2 argumenti: teksta rinda un vesels skaitlis.

Metode salīdzina teksta rindas garumu ar doto skaitli // pārbaudīt vai teksta rindas "Hello World" garums ir 11.

- soli analogiski. izveidojam metodi, tās atgriežamais tips ir boolean. divi argumenti, pirmais ir String bet otrs int.

- Atceramies ka String mainīgām garumu pārbauda ar length() metodi. piemēram:

```
if (strArg.length() == intArg) ...
```

- izveidot metodi test6() kura atgriež bool vērtību, Metodei ir 2 String argumenti. Pārbaudīt vai argumentu teksta rindas satur vienādu tekstu.

- soļi analogiski. izveidojam metodi, tās atgriežamais tips ir boolean. divi argumenti, pirmais ir String bet otrs Integer.
- atcerieeties ka String klases rādītājus saalīdzina ar equals() metodi. piem

```
String strVal = "Hello";  
if(strVar.equals(anotherStrVal)){ .. }
```

- izveidot metodi test7() kura atgriež veselu skaitli : doti divi veseli int tipa skaitļi, atgriezt lielāko no tiem.
  - soļi analogiski. izveidojam metodi, tās atgriežamais tips ir bool. Divi argumenti, abi int tipa.
- kopēja piezīme:
  - ja int tipa vietā izmanto Integer tad salīdzināšana uz vienādību jāveic ar equals() metodes palīdzību.

```
// Izveidojam klasi  
  
class BoolExpressions {  
  
    void test1(int intArg){  
  
        return (i == 10 || i == 25 || i == 75); // metode atgriež boolean vērtību.  
  
    }  
  
}
```

## Uzdevums 2 - testa izveidošana

---

Soļi

- izveidot klasi BoolExpressionsTest
  - Testa klasi veidojam identiski parastai klasei. Ievietojam pakotnē kurā atrodas testējamā klase.
- katras no BoolExpressions klases metožu testēšanai izveidot atsevišķu testa metodi
- (Neobligāti) - iespējams izveidot divas testa metodes: vienu pozitīvam testam otru negatīvam.
- Izpildiet klasi, pārliecinieties ka visi testi ir PASSED.

piemēram

```
public class BoolExpressionsTest {

    void positiveTestOfMethodTest1() {
        BoolTest app = new BoolTest();
        bool expectedValue = true;
        bool actualValue = app.test1(20); // pozitīvs tests atbilst prasībām
        System.out.println(checkTest(actualValue,expectedValue, "test1 metodes pozitīvs tests"));
    }

    void negativeTestOfMethodTest1() {
        BoolTest app = new BoolTest();
        bool expectedValue = true;
        bool actualValue = !app.test1(1); // negatīvs tests ne/atbilst prasībām
        System.out.println(checkTest(actualValue,expectedValue,"test1 metodes negatīvs tests"));
    }

    // vairāk testu ...

    // kopējā metode PASSED/NOT PASSED izvadām
    String checkTest(bool actualValue, bool expectedValue) {
        if (actualValue==expectedValue) return "PASSED";
        else return "NOT PASSED";
    }
    // kopējā metode PASSED/NOT PASSED izvadām
```

```
String checkTest(String actualValue, String expectedValue) {  
    if (actualValue.equals(expectedValue)) return "PASSED";  
    else return "NOT PASSED";  
}
```

```
} // end of class
```