Funkcionalno programiranje Kolokvijum 2017 Rešenja zadataka

**Pitanja**

1. [10] Koja je osnovna razlika u upotrebi kontrolne strukture uslovnog grananja u jeziku Scala u odnosu na jezik Java?

2. [10] Napisati definiciju pojma *funkcija višeg reda*.

3. [10] Da li primarni konstruktor klase A može da pozove javni pomoćni konstruktor te klase? Obrazložiti.

4. [10] U kakvoj vezi su klase `scala.AnyRef` i `scala.Null`?

5. [10] Date su klase A i B, pri čemu je klasa B direktno ili indirektno izvedena iz klase A. Napisati deklaraciju generičke funkcije, uz ograničenje tipa generičkog parametra S koji istovremeno mora biti podtip tipa A i nadtip tipa B. Funkcija kao parametar prima podatak tipa Int, a kao rezultat vraća funkciju koja podatak tipa S pretvara u podatak tipa Int.

## Zadaci

1. [15] Napisati funkciju koja prima realan broj `x` i ceo broj `n`, i računa vrednost $x^n$ primenom terminalne rekurzije. U slučaju negativne vrednosti `n`, funkcija baca izuzetak tipa `IllegalArgumentException`.

```
def pow(x: Double, n: Int) = {
        @tailrec
        def powi(p: Double, n: Int) : Double =
                if( n == 0 ) p
                else powi(p*x, n-1)

        if( n < 0 ) throw new IllegalArgumentException
        powi(1, n)
}
```

2. [15] Napisati potrebne klase za realizaciju binarnog stabla celih brojeva. Napisati funkciju koja utvrđuje da li dato stablo celih brojeva zadovoljava uslove heap-a, odnosno da li za svaki čvor važi da vrednost smeštena u njemu nije manja od vrednosti u njegovim direktnim potomcima.

```
abstract class BinTree {
  def isHeap : Boolean
  def <(x: Int) : Boolean
  def >(x: Int) : Boolean
  protected def test : Boolean = false
  override def toString = test.toString()
}

class Empty extends BinTree {
  def isHeap = true
  def <(x: Int) = true
  def >(x: Int) = true
  protected override def test = true
}
```

```
class Node(i : Int, l: BinTree, r: BinTree)
  extends BinTree {
    def isHeap = {
        l < i && r > i && l.isHeap && r.isHeap
    }

    def <(x: Int) = i < x
    def >(x: Int) = i > x
    protected override def test = true
}
```

3. [20] Napisati generičku funkciju koja pravi kerifikovanu verziju funkcije sa dva parametra tipa U i V i čiji rezultat je tipa T?

```
def curry[U, V, T](f: (U,V) => T) : U => V => T =
{
        x => y => f(x,y)
}
```