

## Programiranje I: kolokvij

18. november 2019

Čas reševanja je 60 minut. Veliko uspeha!

### 1. naloga

**a)** Definirajte funkcijo `is_root`, ki sprejme dve celi števili in preveri, ali je prvo število kvadratni koren drugega.

```
# is_root 10 100;;  
- : bool = true  
# is_root (-2) 4;;  
- : bool = false
```

**b)** Definirajte funkcijo `pack3`, ki sprejme tri argumente in vrne njihovo trojico.

```
# pack3 1 false [];;  
- : int * bool * 'a list = (1, false, [])
```

**c)** Definirajte funkcijo `sum_if_not : (int -> bool) -> int list -> int`, ki vrne vsoto vseh elementov, ki ne ustrezajo danemu predikatu. Za vse točke mora biti funkcija repno rekurzivna.

```
# sum_if_not ((=) 3) [1;2;3;4;5;-3];;  
- : int = 9
```

**d)** Definirajte funkcijo `apply : ('a -> 'b) list -> 'a list -> 'b list list`, ki vrne seznam seznamov, ki ga dobimo tako, da vse funkcije iz prvega seznama uporabimo na vseh elementih drugega seznama. Za vse točke mora biti funkcija repno rekurzivna.

```
# apply [(+) 1; (-) 2; ( * ) 3] [1; 2; 3];;  
- : int list list = [[2; 1; 3]; [3; 0; 6]; [4; -1; 9]]  
# apply [(<) 1; (=) 2; (>) 3] [1; 2; 3; -1];;  
- : bool list list =  
[[false; false; true];  
 [true;  true;  true];  
 [true; false; false];  
 [false; false; true]]
```

## 2. naloga

Delavnik zaposlenega na fakulteti lahko predstavimo s seznamom srečanj, od katerih je vsako predstavljeno z imeni predmeta, vrsto (predavanja/vaje) ter številom šolskih ur. Urnik potem enostavno predstavimo kot seznam seznamov srečanj (pri čemer privzamemo teden na FMF-ju ima lahko poljubno število dni).

**a)** Definirajte:

- variantni tip `vrsta_srecanja` s konstruktorjema `Predavanja` in `Vaje`,
- zapisni tip `srecanje` s polji `predmet`, `vrsta` ter `trajanje`,
- tip `urnik` kot okrajšavo za `srecanje list list`.

**b)** Definirajte primera srečanj `vaje`, ki predstavlja tri ure vaj pri Analizi 2a, ter `predavanja`, ki predstavlja dve uri predavanj pri Programiranju 1.

**c)** Definirajte primer `urnik_profesor` : `urnik` (pomagajte si z že definiranimi vajami in predavanji iz prejšnje naloge), ki ima ob prvi dan dve uri vaj, tretji dan eno uro predavanj, šesti dan ponovno eno uro vaj, preostanek tedna pa ima prosto.

**d)** Definirajte funkcijo `je_preobremenjen` : `urnik -> bool`, ki za podani urnik preveri, da so v vsakem dnevu kvečjemu štiri ure predavanj in štiri ure vaj.

```
# je_preobremenjen urnik_profesor;;  
- : bool = false
```

**e)** Za vsako uro vaj profesor dobi 1 evro, za vsako uro predavanj pa 2 evra. Definirajte funkcijo `bogastvo` : `urnik -> int`, ki sprejme urnik in izračuna plačilo za profesorja. Funkcija naj bo repno rekurzivna.

```
# bogastvo urnik_profesor;;  
- : int = 5
```