

Programiranje I: kolokvij

18. november 2019

Čas reševanja je 60 minut. Veliko uspeha!

V nalogah si lahko pomagata z rešitvami prejšnjih nalog.

1. naloga

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

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

b) Definirajte funkcijo ki sprejme tri argumente in vrne njihovo trojico.

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

c) Definirajte funkcijo, ki sprejme seznam tipa `int list` in predikat tipa `p : int -> bool` in vrne vsoto vseh elementov, za katere predikat `p` vrne `false`. Za vse točke mora biti funkcija repno rekurzivna.

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

d) Definirajte repno rekurzivno funkcijo, ki sprejme seznam funkcij tipa `('a -> 'b) list` in seznam elementov tipa `'a list`. Funkcija naj vrne nov seznam seznamov, kjer so na vsak element vhodnega seznama aplicirane vse funkcije iz seznama funkcij.

```
# apply;;  
- : ('a -> 'b) list -> 'a list -> 'b list list = <fun>  
# 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

a) Želimo modelirati urnik. Definirajte variantni tip `vrsta_srecanja`, ki določa vrsto vnosa v urniku in ima konstruktorja `Predavanje` in `Vaje`. Nato definirajte zapisni tip `srecanje`, ki vsebuje ime (`string`), trajanje v minutah (`int`) in pa vrsto (`vrsta_srecanja`).

b) Definirajte primera za vaje iz programiranja ("Prog1 - Vaje", 120 minut) in predavanja ("Prog1 - Predavanja", 180 minut).

c) Urnik predstavimo kot seznam dnevnih razporedov predmetov (teden na FMF-ju ima lahko poljubno število dni), kjer je dnevni razpored predmetov seznam predmetov za ustrezen dan. Tip urnika bo torej seznam seznamov predmetov oz. (`srecanje list`) `list`.

Definirajte primer `urnik_profesor` (pomagajte si z že definiranimi vajami in predavanji iz prejšnje naloge), ki ima ob prvi dan dvakrat vaje iz programiranja, tretji dan eno predavanje, šesti dan ponovno ene vaje, preostanek tedna pa ima prosto.

d) Definirajte funkcijo, ki za podani urnik preveri, da je v vsakem dnevu kvečjemu 240 minut predavanj in 240 minut vaj.

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

e) Za vsako minuto vaj profesor dobi 1 cent, za vsako minuto predavanj pa 2 centa. Definirajte funkcijo, ki sprejme urnik in izračuna plačilo za profesorja. Funkcija naj bo repno rekurzivna.

```
# bogastvo urnik_profesor;;  
- : int = 720
```