

Komparativna analiza programskih jezika

Laboratorijske vježbe 8

1. Implementirati modul Datum. Omogućiti korisniku da može kreirati samo validne datume (npr. ne može kreirati datum 32.03.1990.). Implementirati sljedeće funkcije:
 - *koliko_dana_između(datum1, datum2)* koja za dva datuma vraća koliko je dana između njih;
 - *uvecaj_datum(datum, broj_dana)* koja proslijeđeni datum uvećava za broj_dana i vraća taj novi datum;
 - *da_li_je_prestupna(datum)* koja za proslijeđeni datum vraća da li je godina prestupna;
 - *datum_to_string(datum)* koja vraća string lijepo formatiranog datuma (npr. "2. Januar 2023")
 - *koji_dan(datum)* koja za proslijeđeni datum vraća koji je dan bio (Ponedjeljak, Utorak, ...) (pogledati Zeller's rule)

Za vježbu:

2. Implementirati modul i modul type (signature) za klasu RGBColor. Svaka boja ima tri atributa *r*, *g* i *b*, koji su cijeli brojevi između 0 i 255. Unutar modula je potrebno definisati *type color* za boju. Onemogućiti korisniku da pravi boje sa invalidnim atributima. Pored toga modul treba da sadrži sljedeće:
 - *red: color*: predstavlja boju $r=255, g=0, b=0$ (**public**)
 - *green: color*: predstavlja boju $r=0, g=255, b=0$ (**public**)
 - *blue: color*: predstavlja boju $r=0, g=0, b=255$ (**public**)
 - *max_255: int -> int*: funkcija koja vraća proslijeđeni parametar ako je on manji ili jednak od 255, u suprotnom vraća 255 (**private**)
 - *add: color -> color -> color*: funkcija koja sabira dvije boje, pri čemu ako zbir neka dva atributa prelazi 255, uzima se vrijednost 255 za taj atribut (**public**)
 - *darken: int -> color -> color*: funkcija koja dijeli svaki atribut boje sa proslijeđenim prirodnim brojem (**public**)
 - *lighten: int -> color -> color*: funkcija koja množi svaki atribut boje sa proslijeđenim prirodnim brojem, pri čemu se ponovo uzima 255 za vrijednost atributa ukoliko ona pređe 255 (**public**)
 - *mix_colors: (color * int * (int->color->color)) list -> color*: funkcija koja prima listu čiji elementi sadrže boju, prirodan broj i funkciju *darken* ili *brighten* te vraća boju koja se dobije kada se prvo primijene *darken* ili *brighten* funkcije sa odgovarajućim parametrima na sve boje u listi, a zatim se one saberu. Npr. ako je lista $[(c1, i1, darken), (c2, i2, brighten)]$ rezultat će biti *add (darken i1 c1, brighten i2 c2)*. Obavezno je koristiti *List.fold_left* funkciju za realizaciju tražene funkcije. Demonstrirati poziv ove funkcije van modula (**public**)
 - demonstrirati kako napraviti binding *purple van modula* koji predstavlja boju $r=127, g=0, b=127$.