



FUNKCIONALNO PROGRAMIRANJE

(MATERIJALI PREUZETI OD DOC. DR MILANE
GRBIĆ)

Uvodna predavanja i vježbe



Opšte informacije:

- Prof. dr Aleksandar Kartelj (predavanja i vežbe)
 - <https://www.matf.bg.ac.rs/~aleksandar.kartelj>
 - aleksandar.kartelj@gmail.com (aleksandar.kartelj@matf.bg.ac.rs)
- Neki materijali na sajtu:
<https://poincare.matf.bg.ac.rs/~aleksandar.kartelj/nastava/BLFP2023/>
- Svi materijali na Google classroom

Obaveze studenata

- Predavanja
- Vježbe
- Domaći zadaci
- Seminarski rad/Projekat – 40 poena
- Praktični ispit – 30 poena
- Teorijski ispit – 30 poena

Literatura:

- Introduction to functional programming (John Harrison)
- Programming in Haskell (Graham Hutton)
- Funkcionalno programiranje – prednosti i nedostaci (Nenad Mitić)
- ostalo

Sadržaj kursa:

- Teorijske osnove funkcionalnog programiranja:
 - Lambda račun;
 - Kombinatori;
 - Tipovi;
- Haskell:
 - Tipovi i klase
 - Definisanje funkcija
 - Zadavanje listi
 - Rekurzivne funkcije
 - Funkcije višeg reda
 - Funkcionalni parseri
 - Deklaracija tipova i klasa
 - Lijeno izračunavanje
 - ...



FUNKCIONALNO PROGRAMIRANJE

Uvod

Uvod

- Programi u tradicionalnim programskim jezicima se „oslanjaju“ na promjenu stanja promjenljivih.
- Apstrakcija?

$$\sigma = \sigma_0 \rightarrow \sigma_1 \rightarrow \sigma_2 \rightarrow \dots \rightarrow \sigma_n = \sigma'$$

- Naredbe dodjele, uslovne naredbe, naredbe ponavljanja, funkcije;
- Redoslijed izvršavanja;
- Imperativna ili proceduralna programska paradigma;
- Funkcionalna programska paradigma;
 - Izrazi i izvršenje izraza; Evaluacija izraza;

Uvod

- Imperativni programski jezici
 - Finalno stanje je funkcija ulaznog stanja; $\sigma' = f(\sigma)$.
- Funkcionalni programski jezici
 - Program je izraz koji odgovara matematičkoj funkciji f .
- Funkcionalna vs imperativna paradigma;
 - Nema varijabli, petlji, redoslijeda izvršavanja;
 - Funkcije se tretiraju kao objekti;

ML

```
let rec fact n =  
  if n = 0 then 1  
  else n * fact(n - 1);;
```

C

```
int fact(int n)  
{ int x = 1;  
  while (n > 0)  
  { x = x * n;  
    n = n - 1;  
  }  
  return x;  
}
```


Koncepti funkcionalnog programiranja

- Programiranje i programski jezici;
 - Proceduralni jezici;
 - Instrukcije koje mijenjaju sadržaj memorijskih lokacija;
 - Opisati šta treba izračunati.
 - Organizovati proces izračunavanja u manjim koracima.
 - Organizovati upravljanje memorijom tokom izračunavanja.
 - Podjela poslova;

$((m + n) * (abs(m - n) + 1))div2$

VS

$x := x + 1$

- Deklarativni (aplikativni) jezici:
 - Funkcionalni jezici;
 - Logički jezici;

Koncepti funkcionalnog programiranja

- Metoda izračunavanja (C (Java) vs Haskell);

U C-u (ili JAVA-i):

```
ukupno = 0;
for (i=1; i<=10; i++)
{
    ukupno = ukupno+i;
}
```

U Haskell-u:

```
sum [1..10]
```

Kratak istorijat funkcionalnog programiranja

1.2 Kratak istorijat funkcionalnog programiranja

1. λ račun - Alonzo Church (radovi iz 1932-1933 i 1941)
2. kombinatorna logika - Schönfinkel, Curry i Feys (radovi iz 1924 i 1958)
3. Lisp - John McCarthy (1959)
4. Iswim i SECD mašina - Peter Landin (1964 i 1966)
5. FP - John Backus (1978)
6. ML - Robin Limner (sredina 1970-tih)
7. Različiti funkcionalni programski jezici (SASL, KRC, Miranda) - David Turner (kraj 1970-tih i početak 1980-tih)
8. Haskell (1988, 1998)

Neke karakteristike i mogućnosti modernih funkcionalnih programskih jezika

- 1) Funkcije su osnovne komponente programa. Program predstavlja kompoziciju funkcija.
- 2) Rekurzija je jedan od osnovnih mehanizama izračunavanja.
- 3) Referencijalna transparentnost (Jednake vrijednosti mogu međusobno da se zamjenjuju).
- 4) Mogućnost korištenja funkcija višeg reda;
 - Argument, rezultat, komponenta strukture podataka.
- 5) Stroga tipiziranost jezika. Statička provjera tipova. Polimorfni sistem tipova.
- 6) Rad sa polimorfnim funkcijama.

```
zbir x y = x + y
zbir1 x y = x + y + 1
```

Neke karakteristike i mogućnosti modernih funkcionalnih programskih jezika

7) Mogućnost traženja uzorka (engl. *pattern matching*);

- poklapanje tipova;
- pokrivanje svih mogućih slučajeva;
- nedvosmislenost (od vrha ka dnu);

8) Upotreba Karijevih funkcija;

9) Lijeno izračunavanje;

10) Korisnički definisani tipovi podataka;

11) Rad sa apstraktnim tipovima podataka;

12) Lambda račun i redukcija izraza;

13) Kombinatorni račun;

14) Graf redukcija;

15) ...

```
False && x = False
```

```
True && x = x
```

```
False || x = x
```

```
True || x = True
```

```
not True = False
```

```
not False = True
```

Ekspresivnost funkcionalnih jezika

- Ekspresivnost funkcionalnih jezika je ekvivalentna ekspresivnosti lambda računa.
- Ekspresivnost lambda računa je ekvivalentna ekspresivnosti Tjuringovih mašina (1937).
- Ekspresivnost Tjuringovih mašina je ekvivalentna ekspresivnosti imperativnih jezika.
- Zaključak?

Prednosti i nedostaci funkcionalne programske paradigme

- Tradicionalni programski jezici su (uglavnom) apstrakcija mašine;
- Matematički objekti;

ML

```
let rec fact n =  
  if n = 0 then 1  
  else n * fact(n - 1);;
```

Math

$$\llbracket \text{fact} \rrbracket (n) = \begin{cases} n! & \text{if } n \geq 0 \\ \perp & \text{otherwise} \end{cases}$$

- Funkcije u C-u vs funkcije u matematici;

```
int rand(void)  
{ static int n = 0;  
  return n = 2147001325 * n + 715136305;  
}
```