

FUNKCIJSKI ZNANSTVENI KALKULATOR

Lucija Josipović Deranja, Marko Papić

Sveučilište Jurja Dobrile u Puli

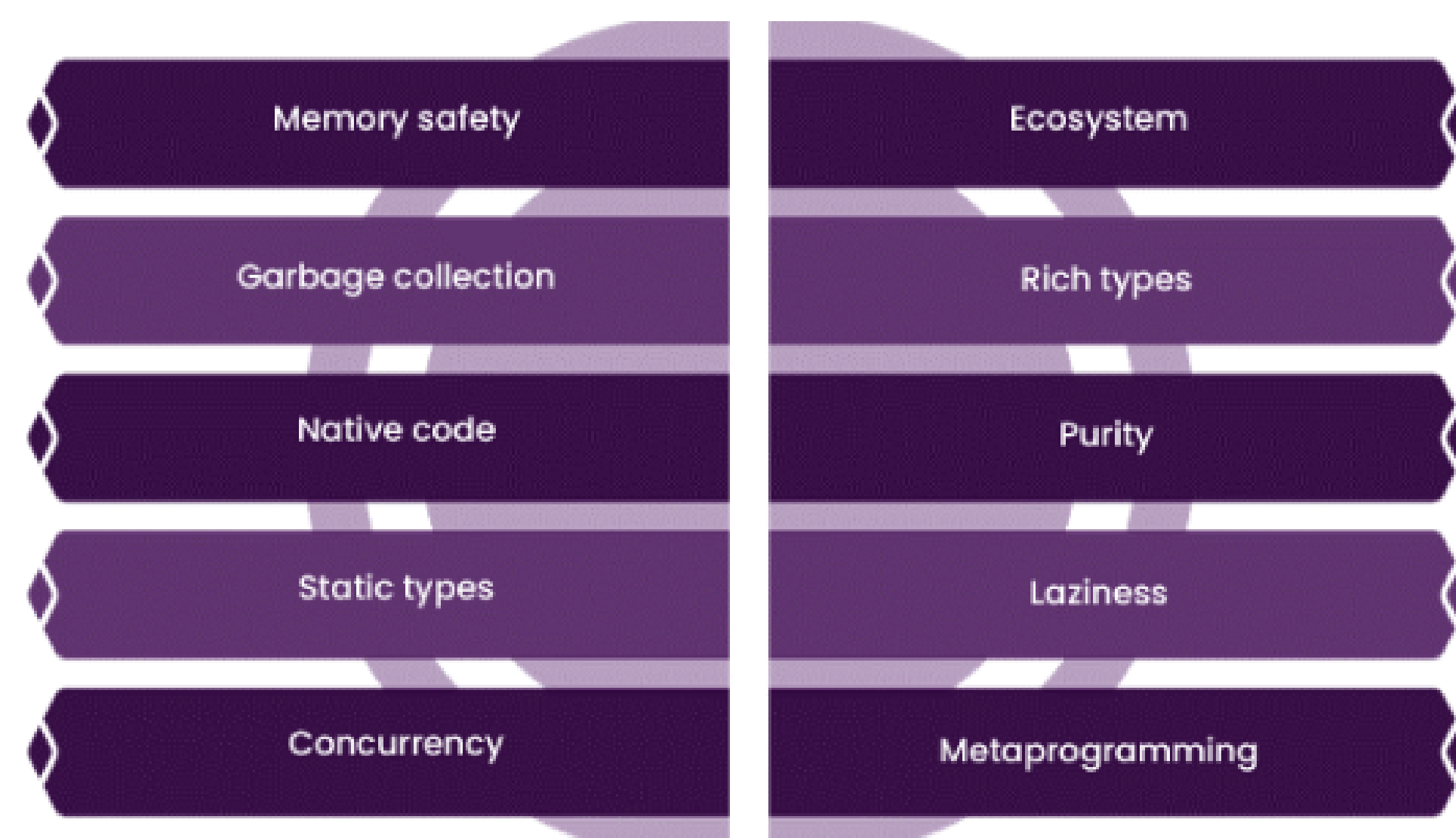


Uvod

Funkcijsko programiranje je paradigma koja se posljednjih godina sve više koristi u razvoju softvera zbog svojih jedinstvenih karakteristika koje omogućuju jednostavnije i efikasnije upravljanje složenim operacijama. Jedan od jezika koji se često koristi za funkcijsko programiranje je Haskell. Haskell je čist funkcijski jezik što znači da sve funkcije u Haskell-u ne mijenjaju stanje i nemaju nuspojava čime se osigurava predvidljivost i jednostavnost razumijevanja koda.

O Haskellu

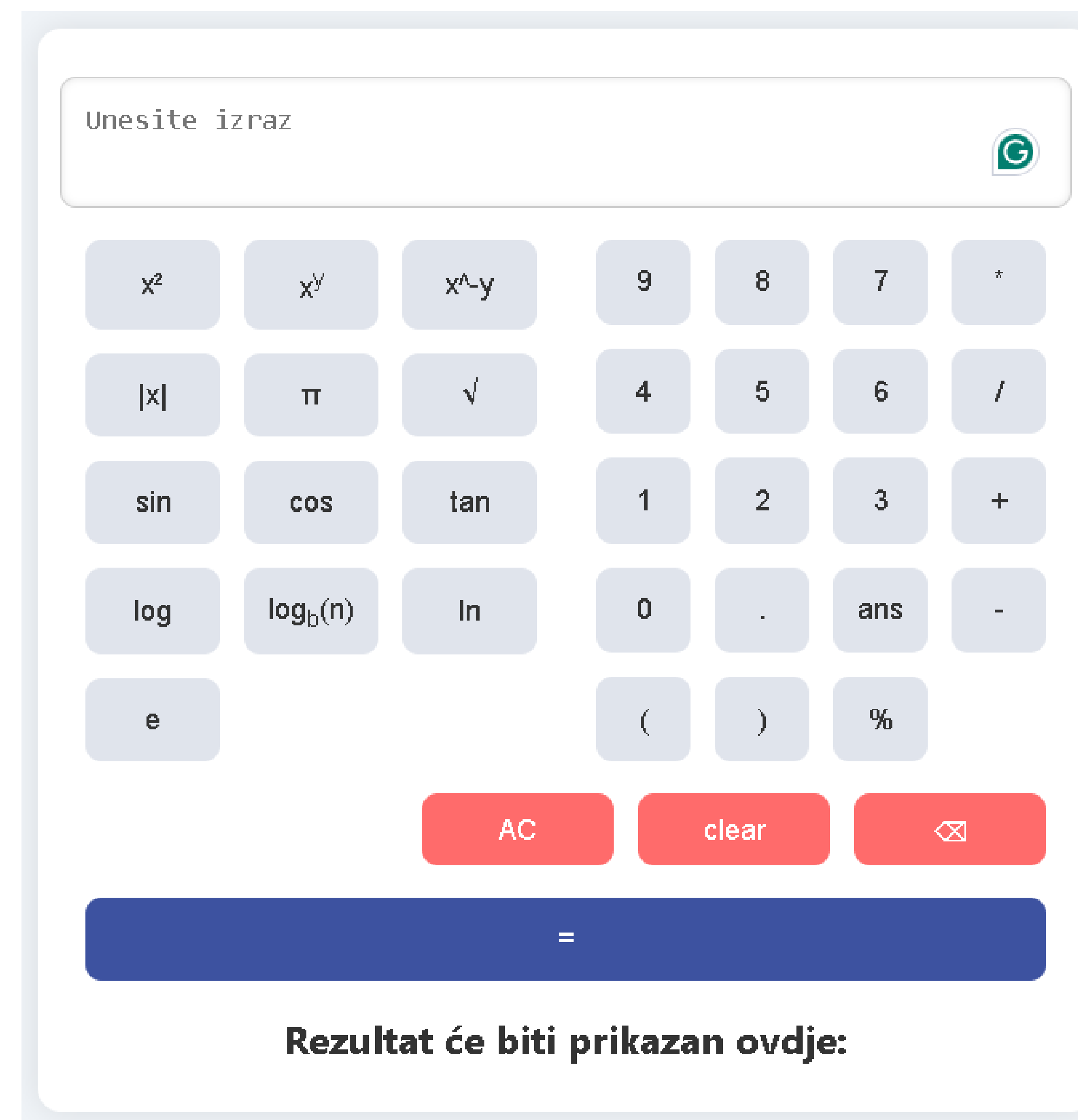
Haskell je funkcijski programski jezik koji se ističe svojom čistoćom, što znači da funkcije u Haskell-u ne mijenjaju stanje i nemaju nuspojava. Razvijen je 1990-ih godina kao zajednički napor istraživača funkcijskog programiranja da stvore standardizirani jezik koji bi ujedinio različite funkcijske jezike tog vremena. Haskell koristi li-jeno evaluiranje, što znači da se izrazi evaluiraju samo kada su potrebni, što omogućuje efikasnije upravljanje resursima i optimizaciju performansi. Jedna od ključnih karakteristika Haskell-a je njegova snažna statička tipizacija, koja omogućuje rano otkrivanje grešaka i osigurava sigurnost koda. Haskell je posebno popularan u akademskim krugovima i istraživačkim projektima, ali se također koristi u industriji za razvoj složenih i kritičnih softverskih sistema.



Arhitektura kalkulatora

Arhitektura funkcijskog znanstvenog kalkulatora uključuje nekoliko ključnih komponenti koje omogućuju efikasnu obradu matematičkih izraza. Program je podijeljen u module koji svaki obavljaju specifičan zadatak:

- Korisničko sučelje: GUI je kreiran koristeći Threepenny biblioteku. Sadrži tekstualno polje za unos izraza
- Dugmad za različite matematičke operacije i dugme za izračunavanje rezultata.
- Parser i evaluacija izraza: Implementiran je parser koristeći Parsec biblioteku za parsiranje matematičkih izraza.

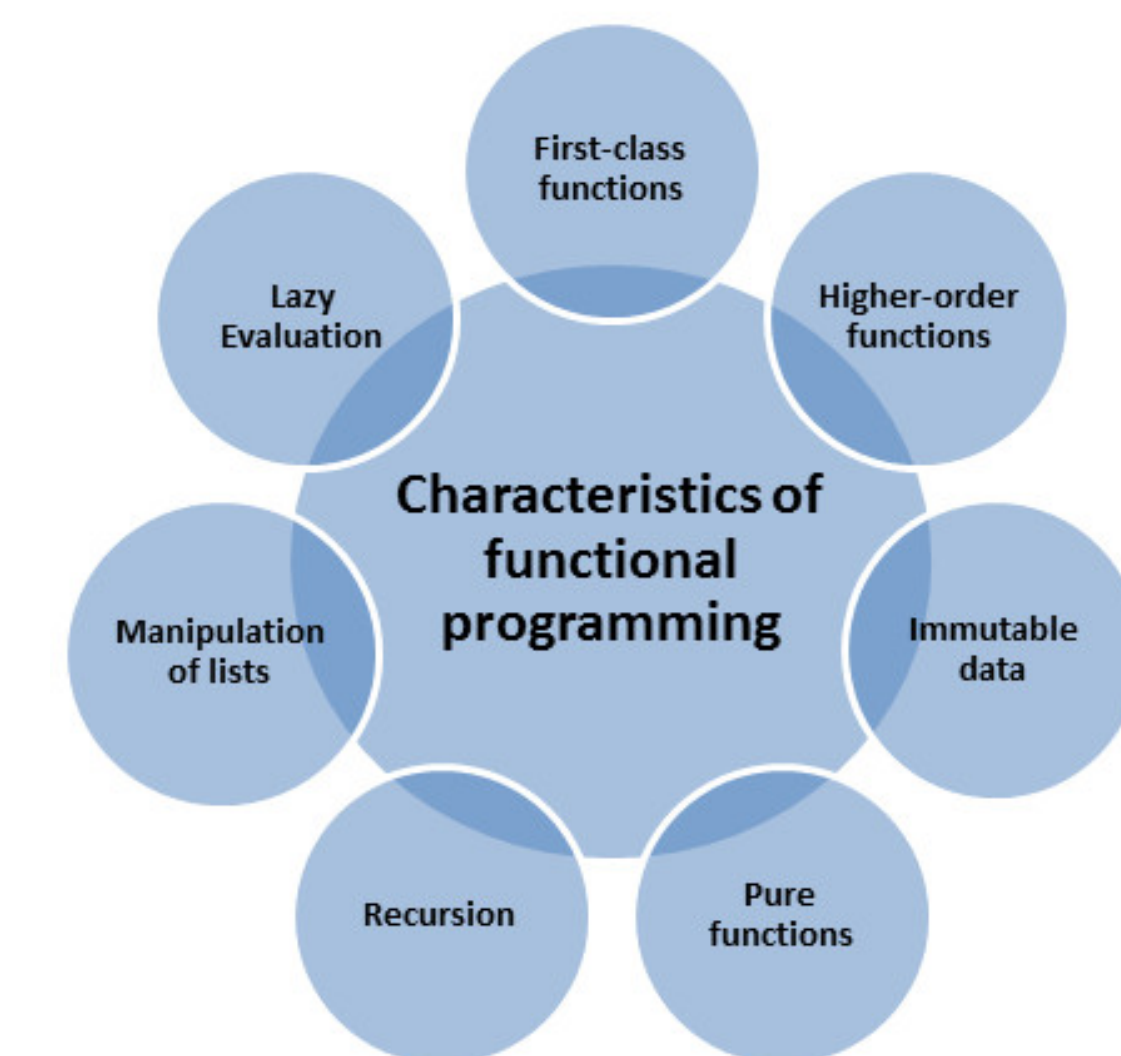


Ključni funkcijski koncepti

Funkcijsko programiranje donosi prednosti poput čistih funkcija, nemjenjivih podataka i kompozicije funkcija. Čiste funkcije osiguravaju predvidljivost i jednostavnost testiranja, olakšavaju paralelizaciju i povećavaju sigurnost koda.

Nemjenjivi podaci olakšavaju praćenje stanja i paralelizaciju, te smanjuju rizik od grešaka.

Kompozicija funkcija omogućuje modularnost i reusabilnost koda, što olakšava razvoj i održavanje sustava.



Prednosti korištenja funkcijskog programiranja

Funkcijsko programiranje nudi nekoliko ključnih prednosti u usporedbi s imperativnim i objektno orijentiranim pristupima. Među najvažnijim prednostima su imutabilnost podataka, čiste funkcije, visokamodularnost i izražajnost.

Zaključak

Funkcijski znanstveni kalkulator u Haskell-u predstavlja primjer kako funkcijsko programiranje može olakšati upravljanje složenim operacijama i održavanje koda. Kroz ovaj projekt demonstrirane su ključne prednosti funkcijskog pristupa, uključujući modularnost, testabilnost i predvidljivost. Dok imperativni pristup može biti učinkovitiji u nekim slučajevima, funkcijski pristup nudi jedinstvene prednosti koje mogu značajno unaprijediti proces razvoja softvera. Razvoj kalkulatora pokazuje kako se složeni problemi mogu elegantno riješiti korištenjem funkcijskog pristupa.