

Uvod
Istorijat
Primena i mogucnosti
Osobine i specificnosti jezika
Funcionalna paradigma - Pattern matching
Asinhrono i paralelno programiranje
Radni okvir - .NET framework
Instalacija i pokretanje
Fizz Buzz
Jedinica mere
Literatura

F# na .NET platformi

T.Todorov T.Garibovic D.Nedeljkovic M.Vicentijevic

23. april 2019

Uvod
Istorijat
Primena i mogucnosti
Osobine i specificnosti jezika
Funcionalna paradigma - Pattern matching
Asinhrono i paralelno programiranje
Radni okvir - .NET framework
Instalacija i pokretanje
Fizz Buzz
Jedinica mere
Literatura

Outline

- 1 Uvod
- 2 Istorijat
- 3 Primena i mogucnosti
- 4 Osobine i specificnosti jezika
- 5 Funcionalna paradigma - Pattern matching
- 6 Asinhrono i paralelno programiranje
- 7 Radni okvir - .NET framework
- 8 Instalacija i pokretanje
- 9 Fizz Buzz
- 10 Jedinica mere
- 11 Literatura

Uvod
Istorijat
Primena i mogucnosti
Osobine i specificnosti jezika
Funcionalna paradigma - Pattern matching
Asinhrono i paralelno programiranje
Radni okvir - .NET framework
Instalacija i pokretanje
Fizz Buzz
Jedinica mere
Literatura

Uvod

Istorijat

Nastao je 1970 god.

Osnova F#-a je ML(*Meta-Language*) jezik baziran na programskom jeziku LISP koji je koren strogo tipizirnih programskih jezika.

U decembru 2001. godine u razmatranje vraća jezik *OCaml* i razvija projekat Caml.NET koji će se kasnije preimenovati u F#. Godine 2002. pojavljuje se prva verzija F# 0.5 koja je bila slabo primećena. *Don Syme* 2004. godine nastavlja intenzivno da razvija ovaj jezik, a početkom 2005. godine izbacuje prvu potpunu verziju F#-a.

Poslednja aktuelna verzija jezika je F# 4.1.

Primena i mogucnosti

F# ima svedenu sintaksu koja omogućava laku citljivost koda i koristi se za resavanje slozenih matematickih algoritama.

F# danas ima siroku primenu u obradi baza podataka, finansijskog modelovanja, statistici i bioinformatici.

F# podrzava sledece paradigme: funkcionalnu, imperativnu, objektno-orjentisanu, paralelnu, distribuiranu, asinhronu, meta programiranje, veb programiranje, skript programiranje, analiticko programiranje...

Sistemi na kojima je podrzan: Linux, MAC, Windows, Android, iOS...

Osobine i specificnosti

Osobine: Specificnosti:

- Bezbedan
- Funkcionalan
- Strogo tipiziran
- Automatski zakljucuje tipove
- Kompatibilan
- Povratna vrednost if/else
- Opciono - return
- Kljucne reci let i mutable
- Pattern matching
- Novi tip option type

Funcionalna paradigma - Pattern matching

- Glavna paradigma programskog jezika F#

Pattern matching

Pattern matching je mehanizam koji koristi dekompoziciju i kontrolu toka podataka za poklapanje obrazaca koriscenjem navedene konstrukcije: **match ... with ...**

```
1 let urlFilter url agent =  
2   match (url,agent) with  
3   | "http://www.control.org", 99 -> true  
4   | "http://www.kaos.org" , _ -> false  
5   | _, 86 -> true  
6   | _ -> false
```

Uvod
Istorijat
Primena i mogucnosti
Osobine i specificnosti jezika
Funcionalna paradigma - Pattern matching
Asinhrono i paralelno programiranje
Radni okvir - .NET framework
Instalacija i pokretanje
Fizz Buzz
Jedinica mere
Literatura

Asinhrono i paralelno programiranje

Radni okvir - .NET framework

Temelj .NET platforme je zajednicka jezicka infrastruktura CLI(*Common Language Infrastructure*).Kodovi se prevode na MSIL(*Microsoft Intermediate Language*) alemblerski jezik. Implementacija MSIL-a na CLI kompajleru je brza i ima sledece prednosti u odnosu na masinski:

- kompatibilnost medju jezicima
- mogucnost rada na vise platformi
- masinska nezavisnost

Mogucnost automatskog prikupljanja smeca je jos jedna prednost. Jos neki okviri: veb radni okviri(*Suave, Fable, ASP.NET Core...*) i radni okviri za testiranje veba (*Web Testing, Frameworks, Unit Testing Libraries...*)

Instalacija i pokretanje

- Alati koji na Windows-u podrzavaju F# se instaliraju u nekoliko koraka:
 - Visual Studio Code
 - Visual studio
 - JetBrains Rider
- Na Linux-u se instalacija vrsi na isti nacin za sledece verzije:
 - Ubuntu
 - Mint
 - Debian

```
1 fsharpc primer.fs
```

Fizz Buzz

```
1 let (|Fizz|Buzz|FizzBuzz|Other|) n =  
2     match (n % 3, n % 5) with  
3     | 0, 0 -> FizzBuzz  
4     | 0, _ -> Fizz  
5     | _, 0 -> Buzz  
6     | _ -> Other n  
7  
8 let fizzBuzz =  
9     function  
10    | Fizz -> "Fizz"  
11    | Buzz -> "Buzz"  
12    | FizzBuzz -> "FizzBuzz"  
13    | Other n -> n.ToString()  
14  
15 seq { 1..100 } |> Seq.map fizzBuzz|> Seq.iter (printfn "%s")
```

Jedinica mere

- Pad orbitera poslatog na Mars 1999.
 - Uzrokovan cinjenicom da je deo softvera koristio numericke, a deo softvera engleske jedinice
- Prevencija gresaka na osnovu konteksta primene
 - Numerickim tipovima se pridruzuju metapodaci
 - Kompajler na osnovu metapodataka proverava ispravnost
- Jedinstveno svojstvo jezika F#
- Primer definisanja jedinice mere

```
1  [<Measure>] type cm  
2  [<Measure>] type inch
```

Jedinica mere

```
1  [<Measure>] type rsd
2  [<Measure>] type eur
3  [<Measure>] type hour
4  [<Measure>] type week
5  [<Measure>] type year
6
7  let hoursBilledPerWeek = 40.0<hour/week>
8  let weeksWorkedPerYear = 35.0<week/year>
9  let rsdPerHour = 1000.0<rsd/hour>
10 let exchangeRate = 0.008547<eur/rsd>
11
12 let eurPerYear = rsdPerHour * hoursBilledPerWeek *
    weeksWorkedPerYear * exchangeRate
13 let bonus = 500.0<eur/year>
14
15 printfn "%f" (eurPerYear + bonus)
```

Uvod
Istorijat
Primena i mogucnosti
Osobine i specificnosti jezika
Funcionalna paradigma - Pattern matching
Asinhrono i paralelno programiranje
Radni okvir - .NET framework
Instalacija i pokretanje
Fizz Buzz
Jedinica mere
Literatura

Literatura