

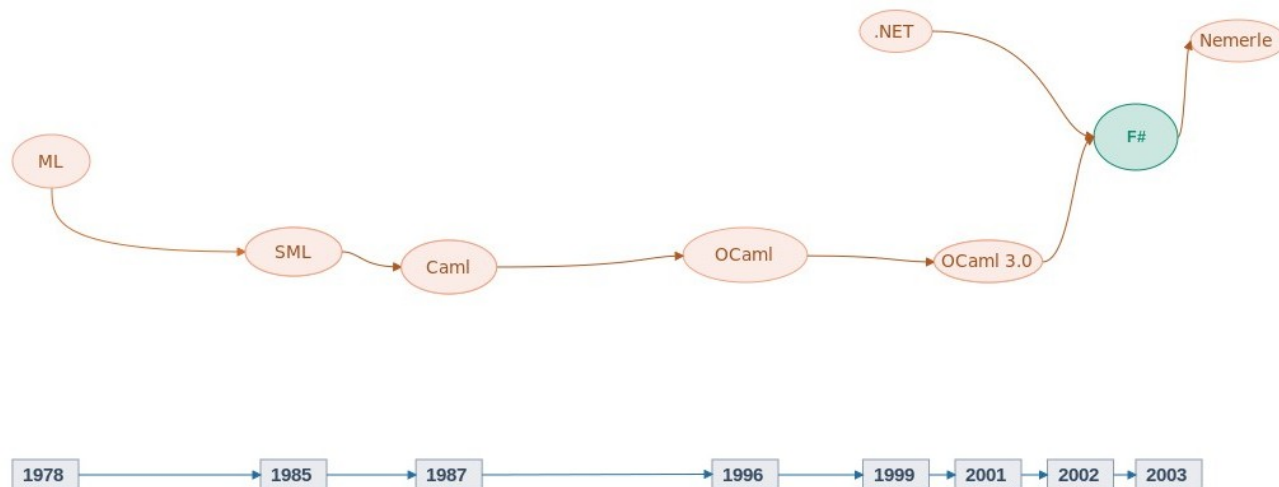
F# na .NET platformi

T.Todorov T.Garibović D.Nedeljković M.Vićentijević

6. maj 2019

- F# je jezik koji danas ima veoma široku upotrebu
- Nastao je iz ideje da se strogo tipizirani funkcionalni jezik preusmeri na .NET platformu
- Autor jezika je Don Syme (eng. *Don Syme*)
- Pretežno je funkcionalan jezik, ali podržava još dosta programskih paradigmi
- F# ima jednostavnu sintaksu, a dovoljno moćnu i za složene matematičke probleme
- Cilj rada je izdvajanje nekih specifičnih karakteristika

Poreklo i uticaj drugih jezika



Slika 1: Razvojno stablo

Primena i mogućnosti

- Jednostavna sintaksa omogućava laku čitljivost koda
- Jezik omogućava brzo generisanje prototipova
- Kôd napisan u F#-u lako može da se paralelizuje
- Jezik ima primenu u oblastima: bioinformatika, baze podataka, statistika, finansijsko modelovanje, ...
- Osim funkcionalnog, jezik F#, podržava još neke tipove programiranja: imperativno, objektono-orjentisano, paralelno, distribuirano, asinhrono, veb programiranje, skript programiranje, ...
- Koristi se na dosta operativnih sistema: Linux, MAC, Windows, Android, iOS, ...

Osobine i specificnosti

Osobine:

- Bezbedan
- Funkcionalan
- Strogo tipiziran
- Automatski zaključuje tipove
- Kompatibilan

Specificnosti:

- Povratna vrednost if/else
- Opciono - return
- Ključne reči 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
```

Asinhrono i paralelno programiranje

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 podržavaju F# se instaliraju u nekoliko koraka:
 - Visual Studio Code
 - Visual studio
 - JetBrains Rider
- Na Linux-u se instalacija vrši na isti način za sledeće 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")
```

- 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)
```

[?]

