Programiranje II

Iztok Savnik, FAMNIT

Februar, 2013.

Vsebina

- Potek predmeta
- Pregled razvoja programskih jezikov
- Koncepti programskih jezikov
- Meta-jezik
- Nekatere primerjave
- Cilji predmeta
- Kaj se dogaja na področju

Potek predmeta

- Predavanja:
  - Predavatelj: Iztok Savnik
- Vaje:
  - Asistenta: Andrej Kramar
- Kratke domače naloge
  - Po vsakem predavanju
  - Ponavljanje snovi
  - 1-2 uri dela
- Dolge domače naloge
  - Dve domači nalogi
  - Izdelava programov

Ocenjevanje

- Sestava ocene:
  - Ocena pisnega izpita – 40%
  - Ocena ustnega izpita – 40%
  - Ocena domačih nalog – 20%
    - Kratke domače naloge – 0%
    - Dolge domače naloge – 20%
- Vsaka od delnih ocen mora biti pozitivna!

Viri

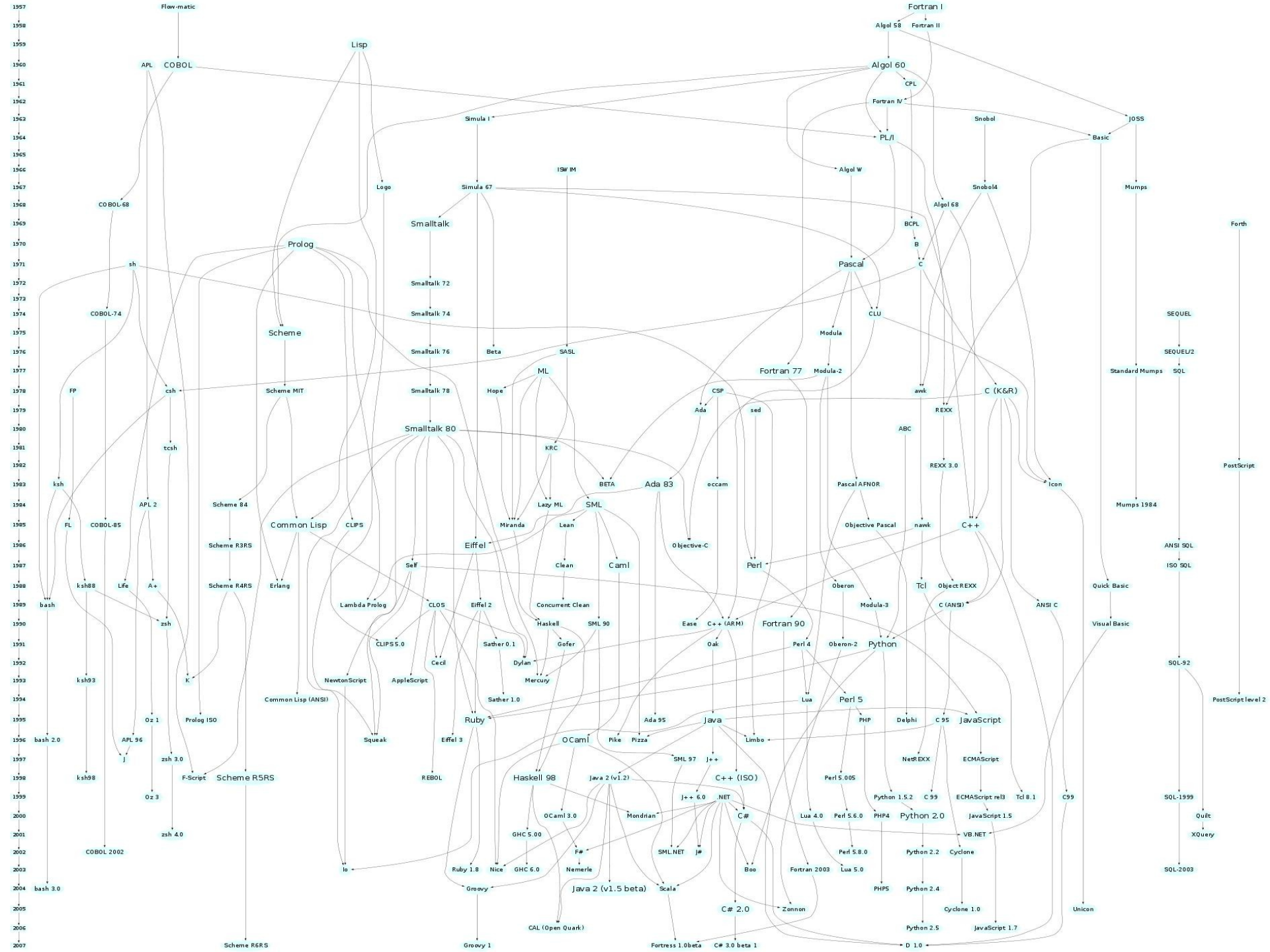
John Mitchell, *Concepts in Programming Languages*, Cambridge Univ Press, 2003.

Iztok Savnik, *Koncepti programskih jezikov*, Skripta, FAMNIT, 2011.

Adam B. Webber, *Modern Programming Languages: A Practical Introduction.* Franklin, Beedle & Associates, Inc., 2003.

Pregled razvoja

- Strojni jeziki
- Fortran
- Cobol
- Algol
  - Programski jezik C
  - Pascal
  - Modula
- Funkcijski jeziki
  - Lisp
  - ML
  - Haskel
- Objektni jeziki
  - C++, Objective C
  - Java, Eifel
  - Ruby
- Visokonivojski jeziki
  - Elektronika
  - Informacijski sistemi
  - Integracijski jeziki
- Skriptni jeziki
  - Sistemski jeziki
  - Web jeziki
  - Uporabniški vmesniki
  - PhP, JSP, ASP
- Prolog
  - Sicstus Prolog
  - SWI-Prolog
  - Datalog
  - Lambda Prolog


1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
Flow-matic
Fortran I
Algol 58
Fortran II
Lisp
APL
COBOL
Algol 60
CPL
Fortran IV
Simula I
Snobol
JOSS
PL/I
Basic
Algol W
ISWIM
Logo
Simula 67
Snobol4
Mumps
Forth
COBOL-68
Algol 68
Smalltalk
BCPL
B
Prolog
C
Pascal
sh
Smalltalk 72
Smalltalk 74
CLU
SEQUEL
Modula
Scheme
SEQUEL/2
Smalltalk 76
Beta
SASL
ML
Fortran 77
Modula-2
Standard Mumps
SQL
FP
csh
Scheme MIT
Smalltalk 78
Hope
CSP
awk
C (K&R)
Ada
sed
REXX
Smalltalk 80
ABC
tcsh
KRC
REXX 3.0
PostScript
ksh
BETA
Ada 83
occam
Pascal AFNOR
Icon
Scheme 84
Lazy ML
SML
Mumps 1984
APL 2
FL
COBOL-85
Common Lisp
CLIPS
Miranda
Lean
Objective Pascal
nawk
C++
Scheme R3RS
Eiffel
Objective-C
ANSI SQL
Self
Clean
Caml
Perl
ISO SQL
ksh88
Life
A+
Scheme R4RS
Erlang
Oberon
Tcl
Object REXX
Quick Basic
bash
Lambda Prolog
CLOS
Eiffel 2
Concurrent Clean
C (ANSI)
Modula-3
ANSI C
zsh
Haskell
SML 90
Ease
C++ (ARM)
Fortran 90
Visual Basic
CLIPS 5.0
Sather 0.1
Gofer
Oak
Perl 4
Oberon-2
Python
ksh93
Cecil
Dylan
SQL-92
K
NewtonScript
AppleScript
Mercury
Common Lisp (ANSI)
Sather 1.0
Lua
Perl 5
PostScript level 2
Oz 1
Prolog ISO
Ruby
Ada 95
Java
PHP
Delphi
C 95
JavaScript
bash 2.0
APL 96
Squeak
Eiffel 3
OCaml
Pike
Pizza
Limbo
J
SML 97
J++
NetREXX
ECMAScript
zsh 3.0
ksh98
REBOL
Haskell 98
Java 2 (v1.2)
C++ (ISO)
Perl 5.005
Tcl 8.1
Oz 3
F-Script
Scheme R5RS
J++ 6.0
.NET
Python 1.5.2
C 99
ECMAScript rel3
C99
SQL-1999
OCaml 3.0
Mondrian
C#
Lua 4.0
Perl 5.6.0
PHP4
Python 2.0
JavaScript 1.5
Quilt
zsh 4.0
GHC 5.00
VB.NET
XQuery
COBOL 2002
F#
Perl 5.8.0
Python 2.2
Cyclone
Io
Ruby 1.8
Nice
GHC 6.0
Nemerle
SML.NET
J#
Boo
Fortran 2003
Lua 5.0
SQL-2003
bash 3.0
Groovy
Java 2 (v1.5 beta)
Scala
PHP5
Python 2.4
C# 2.0
Zonnon
Cyclone 1.0
Unicon
CAL (Open Quark)
Python 2.5
JavaScript 1.7
Scheme R6RS
Groovy 1
Fortress 1.0beta
C# 3.0 beta 1
D 1.0

Strojni jeziki

- Delo z ALU
- Delo z registri
- Ukazi in operandi
- Različna naslavljanja
- Zbirnik
- Makroji
- Procesorji

Fortran

- John W. Backus, 1953, IBM
- The IBM Mathematical **For**mula **Tran**slating System
  - IBM 360
  - Luknjane kartice
- Gradniki Fortran
  - DO, GOTO, IF, SUBRUTINE, CALL
- Zelo popularen jezik za numerično procesiranje
  - Še vedno zelo živ jezik!
  - Implicitni paralelizem
  - Fortran program lahko lahko učinkovito paraleliziramo
  - Edini takšen jezik!
- Fortran II, Fortran 77, Fortran III, Fortran 90, Fortran 95, Fortran 2003

Cobol

- Programski jezik za poslovne aplikacije.
- Grace Murray Hopper, >1950
- Jezik zasnovan po vzoru naravnega jezika.
  - Okorna sintaksa; precej kritik.
  - Cobolski pro<grami delujejo še danes.
  - Sintaksa se je spremenila.

Algol

- Družina programskih jezikov
- Funkcije (procedure)
  - Funcija ima lahko parametre
  - Funkcija lahko vrne rezultat
  - Funkcije si bomo natančno ogledali!
- Jezik ima bloke
  - **`begin … end`**
- Algol, Pascal, C, Delphi, Modula

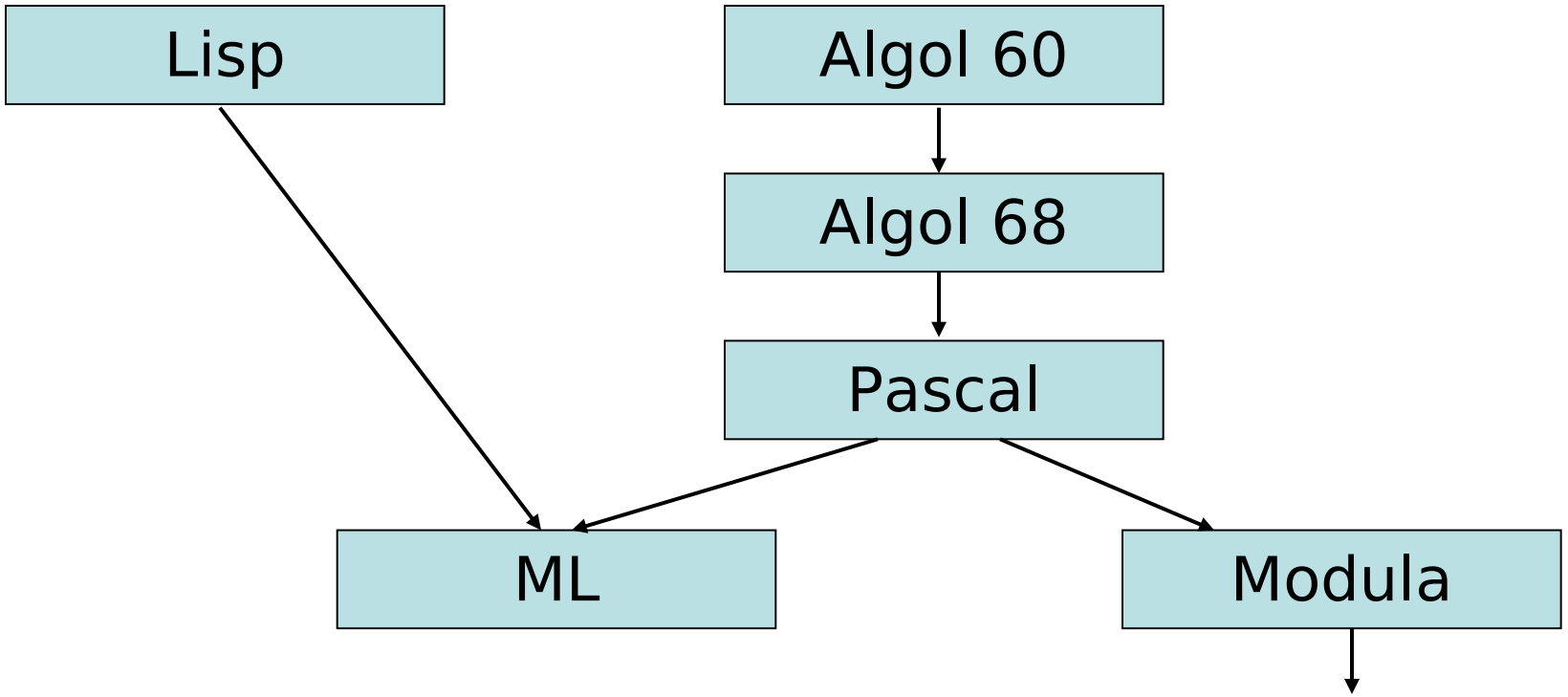
C Programming Language

- Dennis Ritchie, Bell Laboratories, 1972
  - C je narejen za Unix (Unix je C okolje in obratno)
  - Razvit iz jezika B, ki je bil osnovan na BCPL
  - B je bil jezik brez tipov, C dodaja nekatera preverjanja
- Gradniki jezika
  - Strukturiran programski jezik zelo blizu OS in strojni opremi
  - funkcije, operatorji, if, then, for, {}, return, …
  - Še danes najboljši jezik (C++ =? C + iluzija objektov) za implementacijo sistemskih rutin ter resnih in hitrih sistemov
- Polja in kazalci so tesno povezani
  - Polje je preprosto kazalec na prvi element polja
  - E1[E2] =]*((E1)+(E2))
  - Aritmetika za delo s kzalci
- Ritchie je napisal:
  - “C is quirky, flawed, and a tremendous success.”

Pascal

- Niklaus Wirth, ETH Zürich, 1970
- Eden najbolj razširjenih programskih jezikov v Evropi
- Strukturirani programski jezik
  - Procedure, parametri, funkcije
  - Bloki za jezikovne konstrukte
  - Podatkovne strukture, kazalci, polja, variabini zapisi, …
- Veliko različnih implementacij
  - VAX Pascal, Turbo Pascal, …
  - Delphi: zelo lep Objektni Pascal, učinkovito programiranje, uporabniški vmesniki, generiranje kode, …
- Modula-2, Modula-3, Oberon

Sekvence jezikov



Veliko drugih jezikov:

- Algol 58, Algol W, Euclid, EL1, Mesa (PARC), ...
- Modula-2, Oberon, Modula-3 (DEC)

Funkcijski jeziki

- Angleška šola
- Izhajajo iz formalnih jezikov
  - Lambda račun je razvil leta 1930 Alonzo Church
- LISP
  - Razvil John McCarthy
  - Lisp je implementacija λ-računa
- Meta-Language
  - ML, Rob Milner
- Funkcijski jeziki so boljša osnova za učenje?
  - Strogi tipi
  - Program je dokaz
  - Elementi deklarativnega programiranja
  - USA

Zgodovina ML

- Logika za izračunljive funkcije (Robin Milner)
  - Logic for computable functions (LCF)
  - Stanford 1970-71, Edinburgh 1972-1995
- Meta-jezik za LCF sistem
  - Projekt za automatizacijo logike
  - Dokazovanje
  - Sistem tipov
  - Notacija za programe
  - Funkcije višjega reda
  - Programi, ki poiščejo dokaz

Objektno usmerjeni jeziki

- Prvi objektni jezik je bil **`Simula`**.
  - Simulacija dejanskega dogajanja.
- **`Smalltalk`** je eden od najslavnejših jezikov
  - Adele Goldberg, Xerox, Palo Alto
  - Vse na tem svetu so samo objekti!
- Vsak objekt je član razreda
  - Prototipni objekt
  - Razred je objekt!
  - Kompleksne hierarhije dedovanja
  - C++ + Java + vsi novi jeziki !
- Java
  - J.Gosling, B.Joy, G.Steele, G.Bracha
    The Java Language Specification
    http://java.sun.com/docs/books/jls/download/langspec-3.0.pdf

Skriptni jeziki

- Web programski jeziki
  - V zadnjih letih se je pojavila množica jezikov, za delo na Web strežnikih
  - JSP, PhP, JavaScript, ASP, …
  - Uporabljajo vire Web strežnika
- Sistemski programski jeziki
  - Močna povezava z operacijskim sistemom
  - Awk, Perl, Python
  - Zelo učinkovito delo s sistemom
  - Novejši programski gradniki na “sistemski” način
  - Primerni za povezovanje komponent sistema

Visokonivojski jeziki

- Informacijski sistemi
  - 4GL, PL/SQL, SQL3
  - UML (specifikacije)
- Integracijski jeziki
  - Python (multi-paradigm programming language)
  - Perl (OS, Web, Statistika, …)
  - Web jeziki (dostop do vsega…)
- Elektronika
  - VHDL

Prolog

- Programming in Logic, Robert Kowalski
- Predikatni račun
  - Hornovi stavki
  - Unifikacija, resolucija
- Zelo močen, enostaven in abstrakten jezik
- Uporaba vračanja (back-tracking)
  - Deklarativen in proceduralen pomen Prologa
  - Operator CUT (!)
- Jeziki za delo s podatkovnimi bazami
  - Datalog
- Sicstus, SB Prolog, SWI Prolog,…

Koncepti programskih jezikov

- Koncepti programskih jezikov so abstrakcije s katerimi predstavimo strukturo in obnašanje sistemov, ki jih s programom modeliramo.
- Klasifikacije
  - Postopkovne in podatkovne abstrakcije.
  - Jezikovni in implementacijski koncepti.
  - Imperativni in funkcijski jeziki: procedure in funkcije.
  - Objektni jeziki: objekti, razredi.
- Model jezika
  - Konceptualno ogrodje za programiranje.
  - Uporabljene abstrakcije določijo model jezika.

Koncepti programskih jezikov?

- Koncepti vgrajeni v programski jezik !
  - Funkcija, objekt, razred, metoda, abstraktni razred, ...
  - klasifikacija, specializacija, agregacija, ...
- Ločimo med postopkovnimi in podatkovnimi abstrakcijami (meja se briše).
- Model jezika
  - Imperativni, funkcijski, objekti in modularni.
- Ločimo med jezikovnimi in implementacijskimi koncepti
  - Implementacijski koncepti: delo s spominom, imenski prostori, implementacij funkcij, objektov, ...

Abstrakcije

- Abstrakcija je opis koncepta, ki se ne sklicuje na konkretne primere.
- Abstrakcija odvzame del informacij, ki se nanašajo na konkretno.
- Abstrakcija pomeni lahko tudi ekstrakcijo čistih sestavin iz surovega materiala.
- Abstrakcija pomeni tudi kreiranje ideje, ki povzema neko realnost.

Nekatere lastnosti abstrakcij PJ

- Čim bližje strojni opremi: bolj enostavni jeziki.
- Enostavne abstrakcije:
  - Števila, znaki, zanke, rutine, ...
- Višje abstrakcije:
  - Objekti, moduli, ...
- Višji nivo abstrakcij jezika:
  - Bolj usmerjen jezik.
  - Bolj splošne so abstrakcije bolj specifičen jezik dobimo.
- V sredini so splošni programski jeziki:
  - Namembnost zelo široka,
  - Med bolj kompleksnimi računalniškimi jeziki.

Študij konceptov programskih jezikov.

1) Poznavanje izraznih zmožnosti jezikov.
2) Pravilna uporaba danih gradnikov programskih jezikov.
3) Pravilna uporaba uveljavljenih konceptov programiranja.
4) Abstrakcije lahko uporabljamo pri načrtovanju programov.

Programski jeziki

- Konceptualno orodje definirano v obliki jezika s katerim lahko zasnujemo, izrazimo in realiziramo računalniške programe.
  - Programski jeziki so blizu človeškem razmišljanju.
  - „Konceptualno vesolje” v okviru katerega lahko predstavimo svoj sistem.
- Koncepti PJ za katere smo skozi raziskave in poskuse v zadnjih nekaj destletjih ugotovili, da so najbolj primerni za programiranje.
  - Potrebno dobro poznavanje konceptov.
  - Dobro je imeti znanje o formalnem ozadju.
  - Izkušnje pri uporabi konceptov.

Koncepti PJ in abstrakcije

- Študij abstrakcij pomaga pri učenju programskih jezikov.
  - COMMUNICATIONS OF THE ACM April 2007/Vol. 50, No. 4
  - IS ABSTRACTION THE KEY TO COMPUTING?
- Uporabljene abstrakcije določijo model jezika.
  - Imperativni jeziki: spremenljivke, zanke, procedure, ...
  - Funkcijski jeziki: funkcije
  - Objektni jeziki: objekti

Koncepti programskih jezikov

- Koncepti programskih jezikov bodo predstavljeni na sledeč način:
  - Najprej bomo identificirali posamezne koncepte.
  - Splošen opis konceptov z matematičnimi osnovami.
  - Predstavitev konceptov v Ocaml kot referenčnem jeziku.
  - Predstavljeni bodo skupni implementacijski koncepti.
  - Primerjava z realizacijami v drugih programskih jezikih.
    - C, C++, Java, ...

Implementacijski koncepti PJ

- Spoznavamo se tudi z implementacijo gradnikov programskih jezikov.
  - Predstavitev spremenljivk, podatkovnih struktur, itd.
- Delo s spominom.
  - Organizacija spomina.
  - Čiščenje spomina.
- Implementacija predstavljenih konceptov PJ.
  - Funkcije v C, Javi, Ocaml, ...
  - Parametri v C, Javi, Ocaml, ...
  - Razredi v Javi, OCaml, ...

Meta-jezik

- Meta Language (ML) je nastal nenamenoma pri izdelavi programa dokazovanje izjav s pomočjo LCF - Logic of computable functions.
  - LCF je verzija lambda računa.
  - Projekt se je izvajal v skupini za umetno ineteligenco na Univerzi Stanford.
  - Vodja projekta je bil Robin Milner.
  - V okviru LCF je Meta-Language služil kot jezik za zapis dokazov; običajno bi spreminjali vsebino datoteke za izdelavo boljšega dokaza.
  - Pomembno je tudi, da zdaj tvorijo nasledniki LCF drevo znanja o sklepanju s pomočjo računalnikov.

Lambda račun

- Lambda račun je osnoven formalizem za študij programskih jezikov.
  - LCF oz. lambda račun tvori osnovni formalizem za vejo računalništva - teorijo programskih jezikov.
  - Teorija programskih jezikov uporablja sklepanje za izpeljevanje tipov, evaluacijo izrazov, dokazovanje lastnosti programov, itd.
  - Z razširjenim LCF lahko predstavimo statično in dinamično semantiko jezika.
  - Lahko preverimo ključne lastnosti jezikov kot so determinističnost in enoličnost evaluacije, stroge tipe, itd.
  - Lambda račun si bomo ogledali v prvem delovnem poglavju.

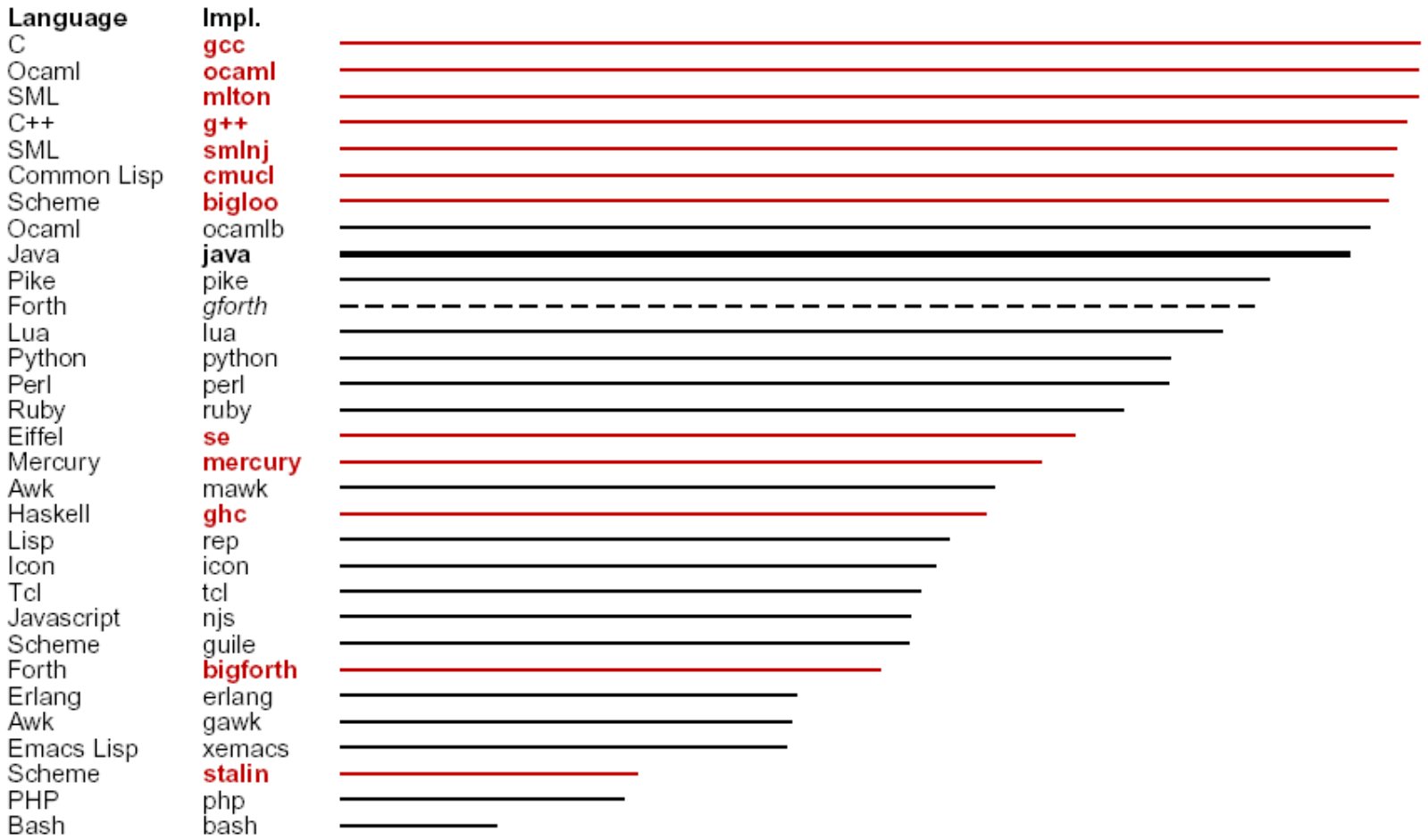
Objektni Caml

- Objektni Caml je referenčni jezik.
  - Jedro Caml je čisti lambda račun.
  - Eden izmed bolje teoretično obdelanih programskih jezikov.
  - Caml ima stroge tipe.
  - Vsebuje imperativne gradnike.
  - Vsebuje parametrični polimorfizem.
  - Vsebuje module in funktorje.
  - Lepa zasnova objektov in razredov.
  - Vsebuje izjeme.
- Omogoča več programskih modelov.
  - Imperativni + funkcijski + objektni programski model.

Nekatere primerjave

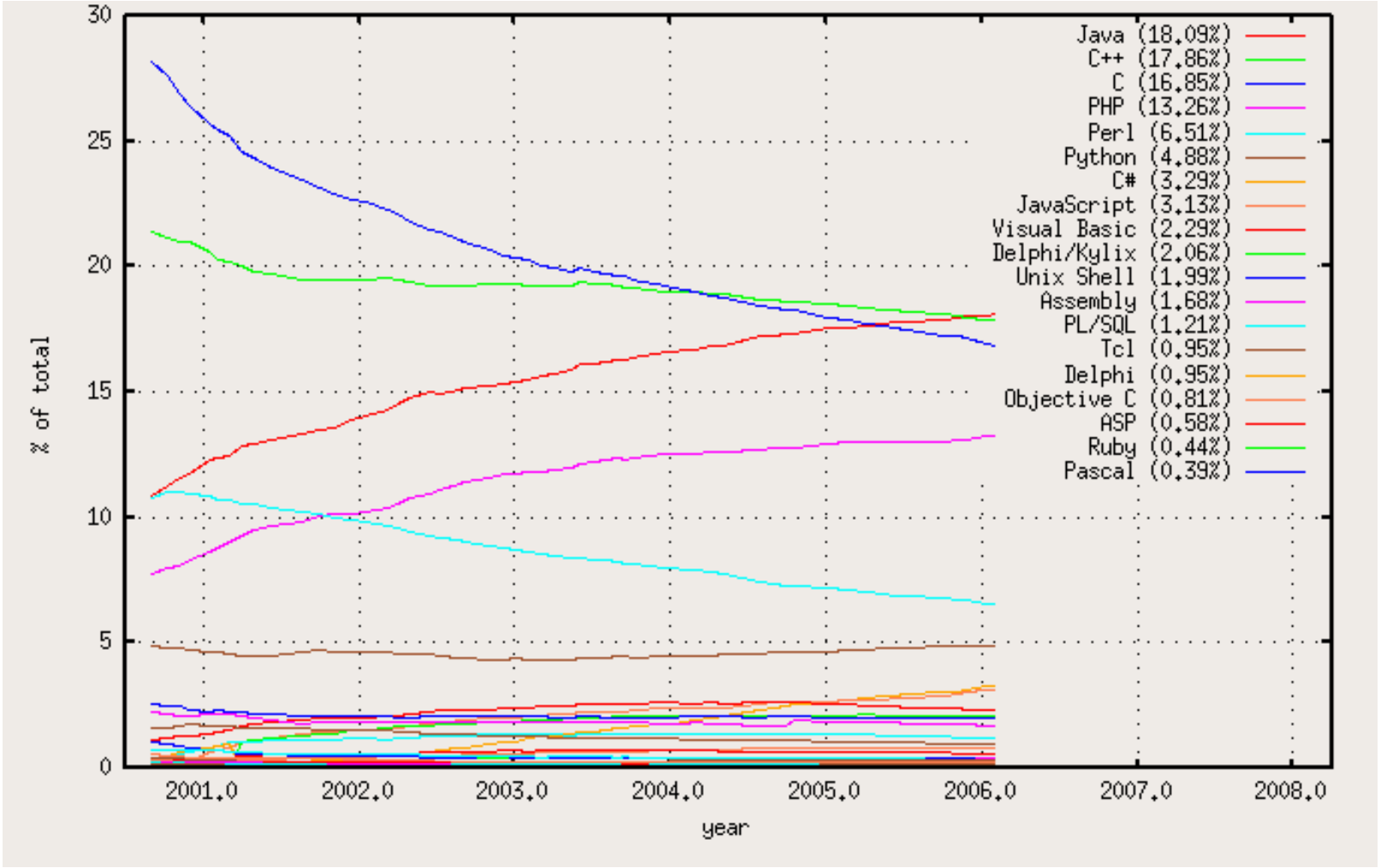
- Hitrost jezikov
- Popularnost jezikov
- The Computer Language Benchmarks Game
- Hitrost, velikost in medsebojna odvisnost PL
- Ocaml for the Masses

Hitrost jezikov



J.Mitchel, predavanja, 2005.

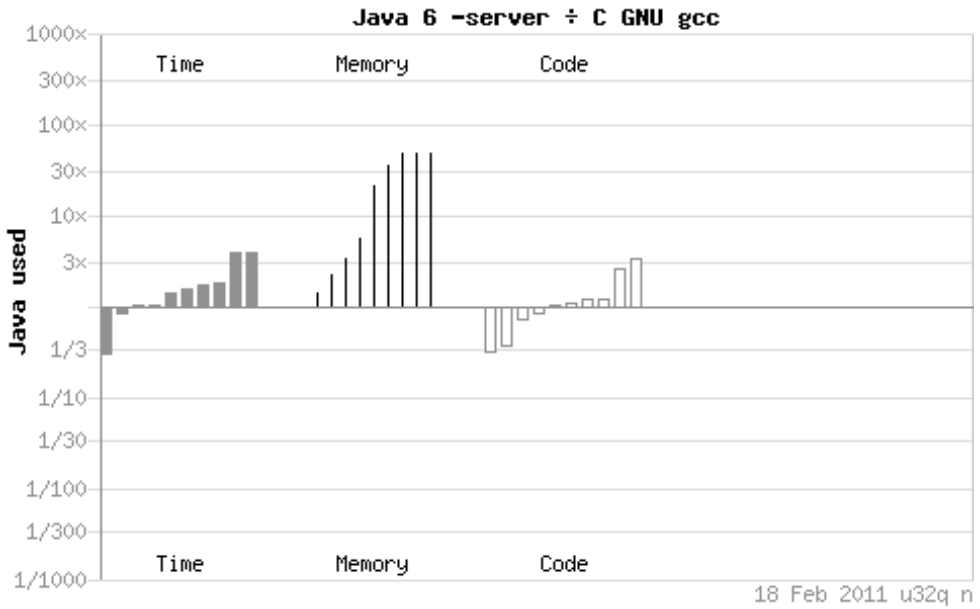
Popularni jeziki

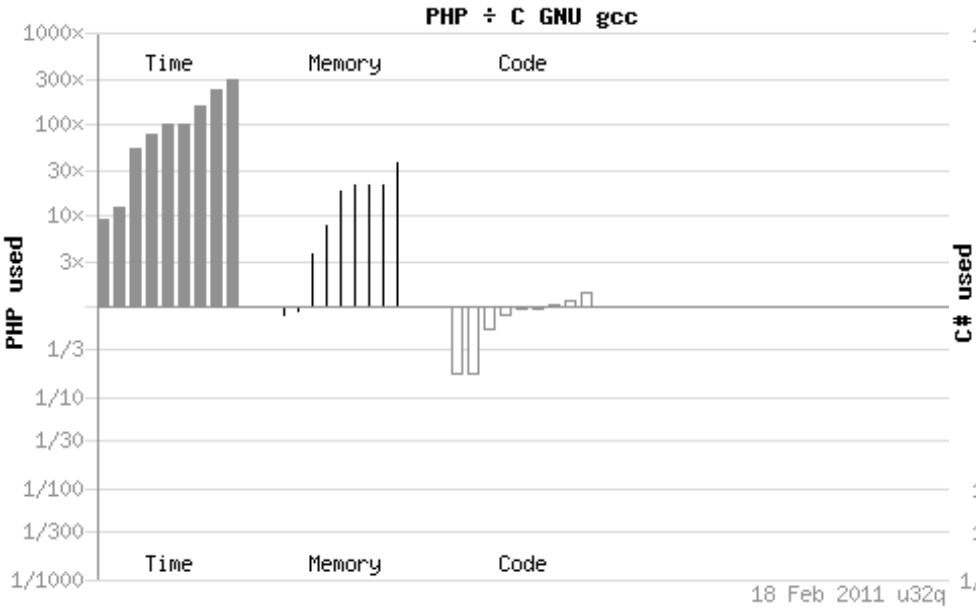


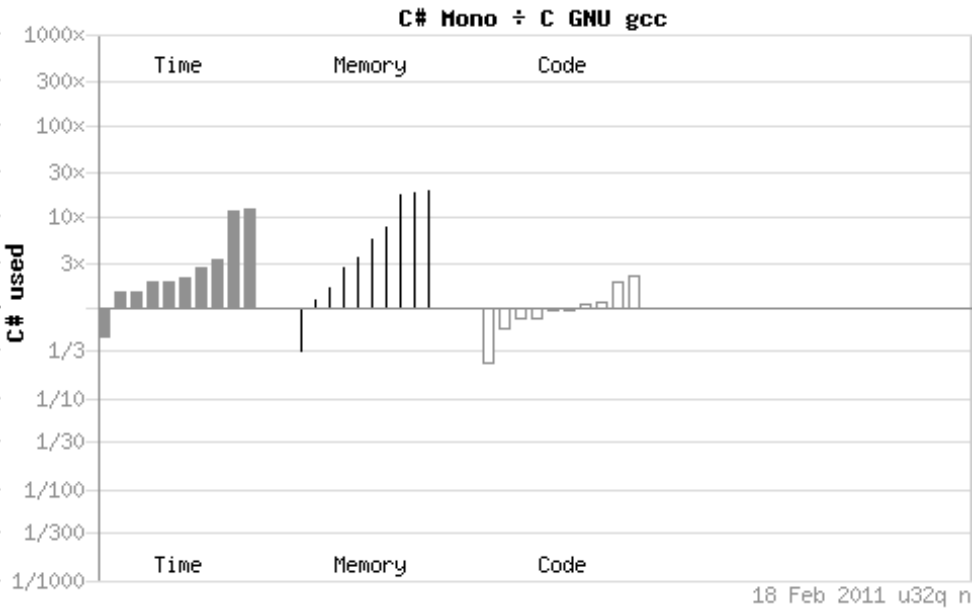
Zbrano: François Labelle, odprtokodni projekti na SourceForge

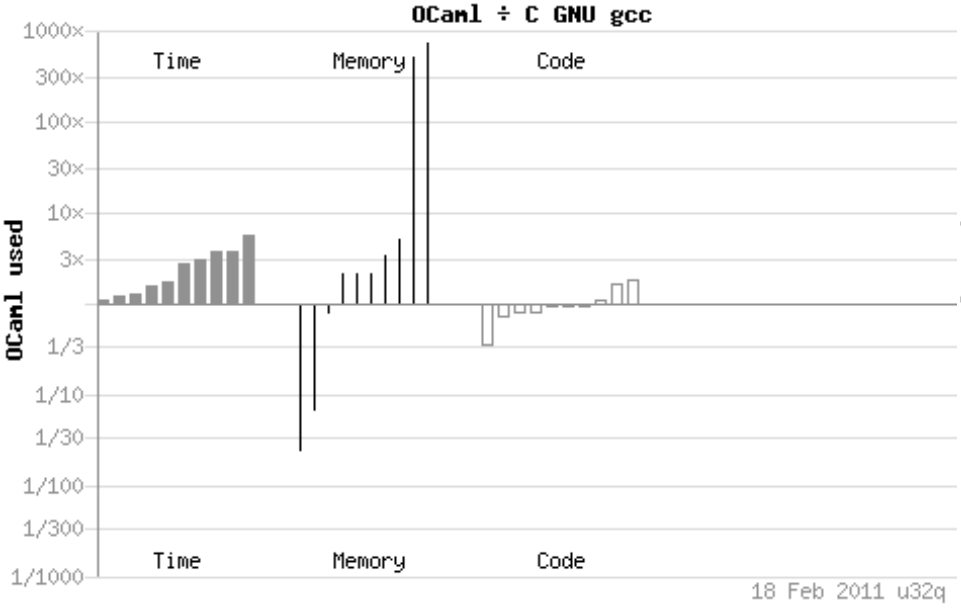
The Computer Language Benchmarks Game

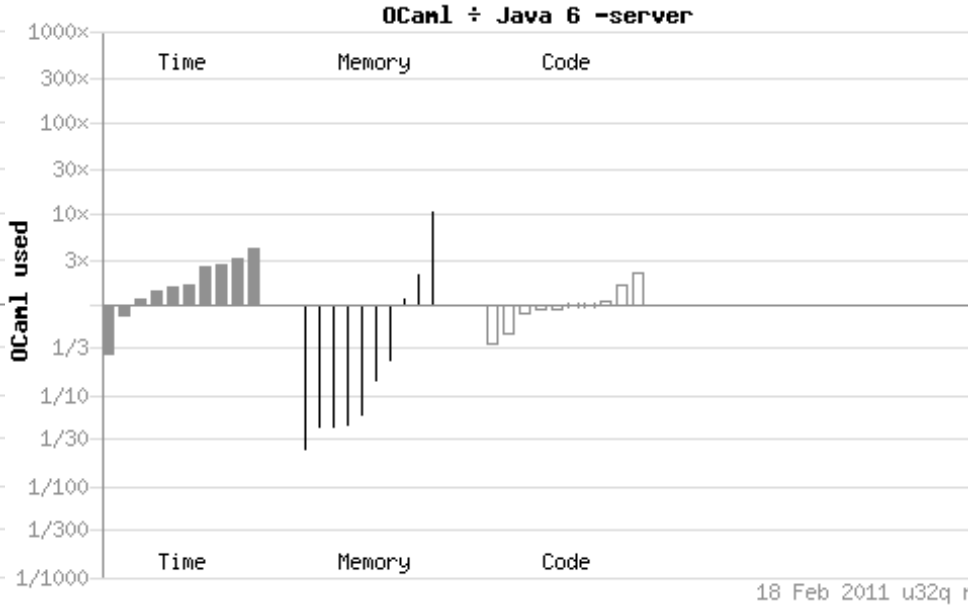
- 24 jezikov, 4 sistemi, 13 testnih programov
- http://shootout.alioth.debian.org/

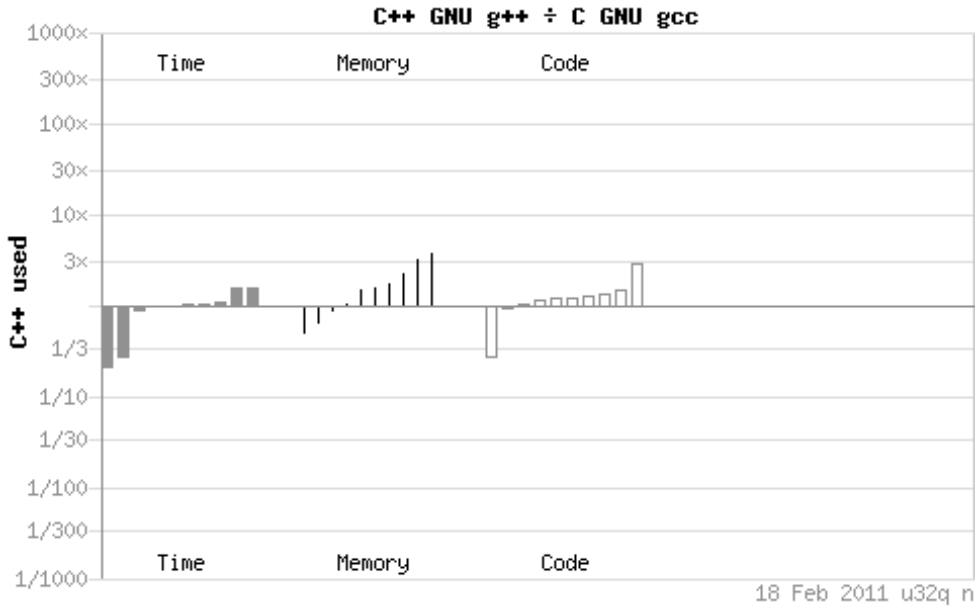



PHP ÷ C GNU gcc
1000×
300×
100×
30×
10×
3×
1/3
1/10
1/30
1/100
1/300
1/1000
PHP used
Time
Memory
Code
Time
Memory
Code
18 Feb 2011 u32q


C# Mono ÷ C GNU gcc
1000×
300×
100×
30×
10×
3×
1/3
1/10
1/30
1/100
1/300
1/1000
C# used
Time
Memory
Code
Time
Memory
Code
18 Feb 2011 u32q n


OCaml ÷ C GNU gcc
1000×
300×
100×
30×
10×
3×
1/3
1/10
1/30
1/100
1/300
1/1000
OCaml used
Time
Memory
Code
Time
Memory
Code
18 Feb 2011 u32q


OCaml ÷ Java 6 -server
1000×
300×
100×
30×
10×
3×
1/3
1/10
1/30
1/100
1/300
1/1000
OCaml used
Time
Memory
Code
Time
Memory
Code
18 Feb 2011 u32q



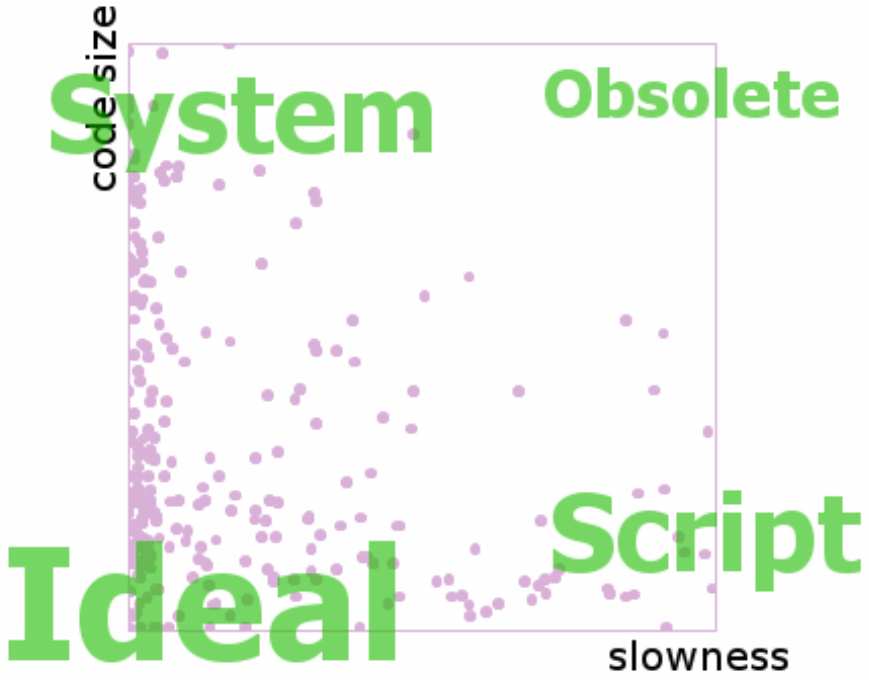
- Višje nivojski prog. jeziki so lahko zelo učinkoviti.
- Java in Ocaml sta zelo učinkovita jezika.

Hitrost, velikost in medsebojna odvisnost PL

429 programov

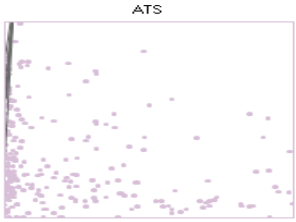
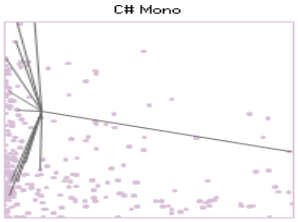
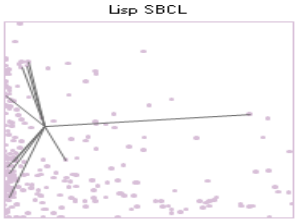
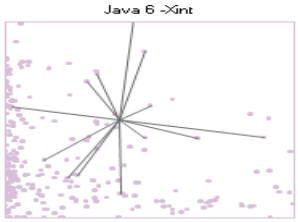
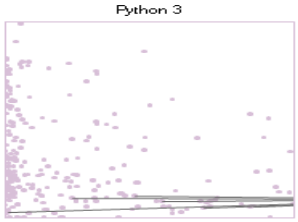
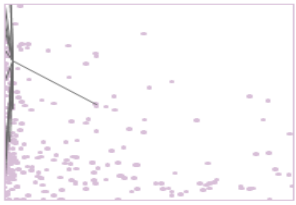
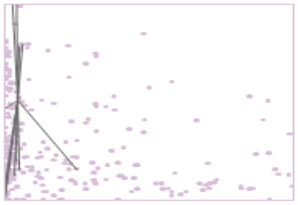
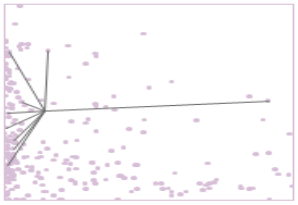
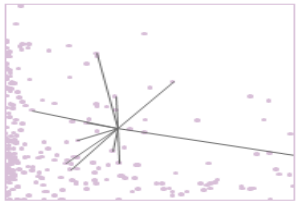
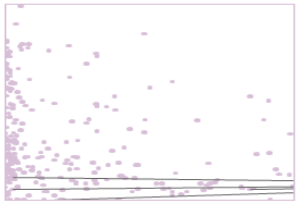
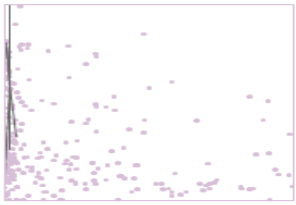
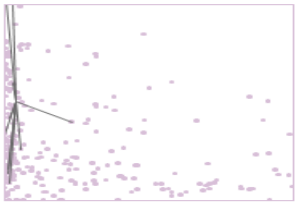
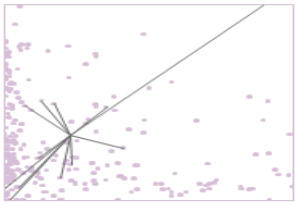
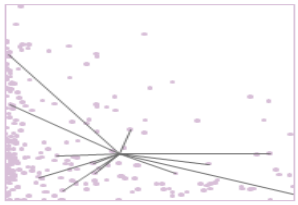
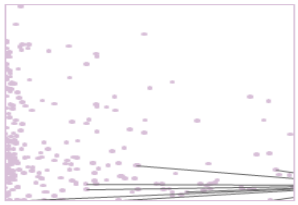
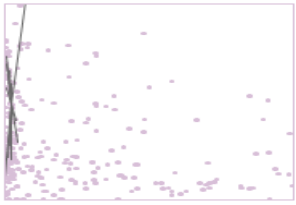
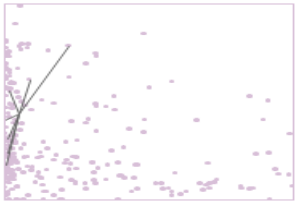
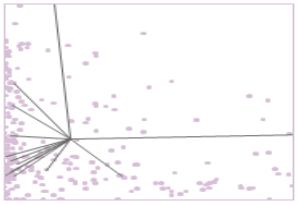
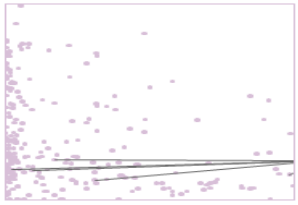
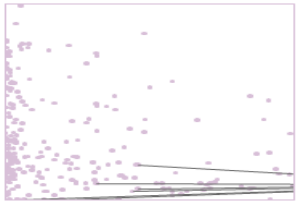
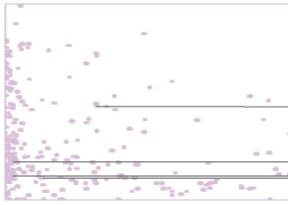
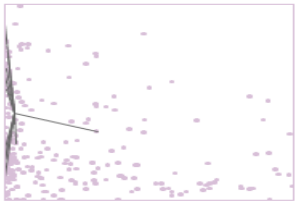
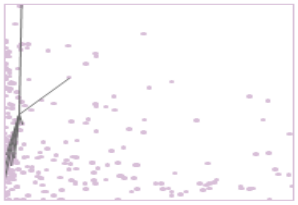
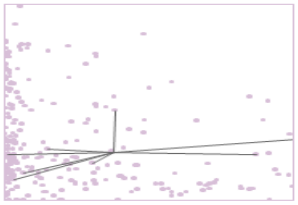
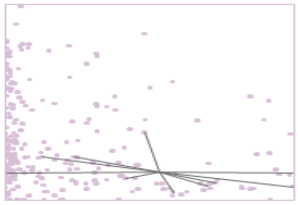
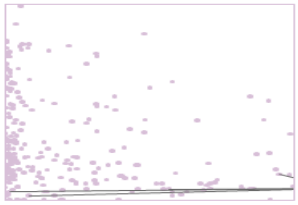
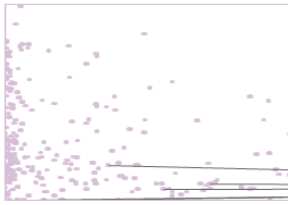
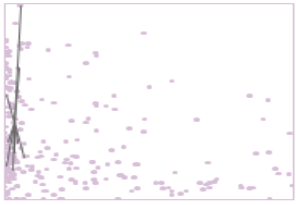
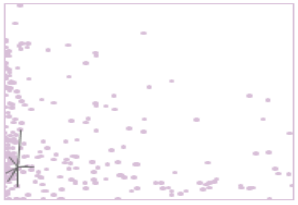
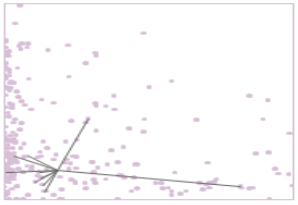
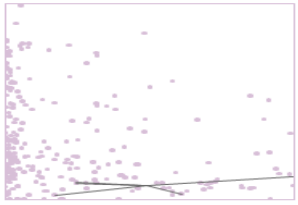
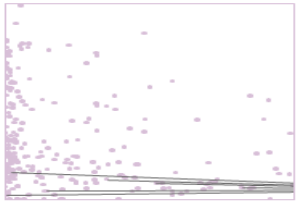
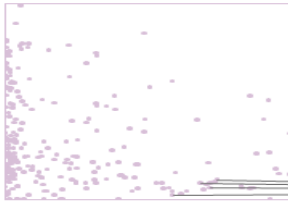
13 testnih programov

33 PL



Eksperiment:

Guillaume Marceau


ATS

C# Mono

Lisp SBCL

Java 6 -Xint

Python 3

C++ GNU g++

Haskell GHC

Ada 2005 GNAT

Smalltalk VisualWorks

Perl

*Java 6 steady state

Pascal Free Pascal

Erlang HiPE

Scheme PLT

Ruby JRuby

Java 6 -server

Lisaac

Scala

Mozart/Oz

Ruby 1.9

PHP

C GNU gcc

Fortran Intel

F# Mono

Lua

JavaScript TraceMonkey

Ruby

Clean

OCaml

Lua LuaJIT

JavaScript V8

Python

Python pypy

communications of the acm
| november 2 0 1 1 | vo l . 5 4 | n o. 1 1

DOI:10.1145/2018396.2018413

Article development led by acmqueue
queue.acm.org

Why the next language you learn should be functional.

BY YARON MINSKY

OCaml for the Masses

Functional programming is an old idea with a distinguished history. Lisp, a functional language inspired by Alonzo Church's lambda calculus, was one of the first programming languages developed at the dawn of the computing age. Statically typed functional languages such as OCaml and Haskell are newer, but their roots go deep—ML, from which they descend, dates back to work by Robin Milner in the early 1970s relating to the pioneering Logic for Computable Functions (LCF) theorem prover.

Cilji predmeta

- Koncepti programskih jezikov
  - Jezik je “konceptualno vesolje” (Perlis)
    - Ogrodje za reševanje problemov
    - Uporabni koncepti in gradniki prorgamskih jezikov
- Spoznavanje več programskih jezikov
  - Spoznavanje z zgodovino pomaga !
  - Konceptov se učimo iz izvorov
  - Vsak jezik ima svoj šarm
  - Učinkovita uporaba gradnikov

Metode

- Lambda račun (Lisp)
  - Močna formalna osnova
  - Osnovni principi proceduralnih jezikov
  - Osnovni principi seznamov
  - Lisp: Eden od prvih PJ je zasnovan na lam.računu
- ML
  - Matematične osnove !!!
  - Osnoven jezik, ki je bi zasnovan za dokazovanje izrekov
  - Strogi tipi
  - Vsebuje večino gradnikov PJ
- Java
  - Primerjalni jezik!!!
  - Eden od bolj razširjenih programski jezik
  - Poudarjen objektni aspekt
  - Strogi tipi, dinamično preverjanje tipov

Metoda

- Primerjava konceptov programskih jezikov
  - Vemo kaj imamo na razpolago
  - Zelo različni gradniki programskih jezikov
- Posamezni koncepti bodo izpostavljeni
  - funkcije, parametri, tipi, območja, kontrola, abstrakcije, dedovanje, polimorfizem, spomin
  - Pogledali si bomo različne pristope
  - Uporabljeni jeziki: C, ML, Java
  - Tudi včasih: Pascal, Fortran, C++, Ada

Metoda

- Spoznavanje z jeziki in implementacijo
  - Vsak gradnik ima svojo ceno
  - Vedno je dobro vedeti kaj se “spodaj” dogaja
  - Učinkovita uporaba gradnikov
- Velikokrat je nujno poznati nekatere aspekte implementacije uporabljenega programskega jezika.
  - Delo s spominom
  - Predstavitev podatkovnih struktur in objektov
  - Implementacija funkcij in rekurzije
  - Dinamični aspekti jezikov

Kaj se dogaja v programskih jezikih

- Komercialni trendi
  - Povečana uporaba jezikov z varnimi tipi
  - Pyton, Java, C#, …
  - Skriptni jeziki
  - Web aplikacije, sistemska okolja, …
  - Sistemska integracija
- Trendi pri učenju
  - Java zamenjuje C in C++
  - Funkcijski jeziki v Evropi
  - ML se je zelo razširil
  - Ruby on Rails
  - Manj je poudarkov na implementacijo podatkov in kontrole

Kaj se dogaja v programskih jezikih

- Raziskovalni trendi
- Programski jeziki za določene namene
  - Mobilne naprave, aparati, procesorji, …
- Omogočeno je vedno več formalne analize
  - Sklepanje s tipi, diagnostika, …
- Metaprogramski jeziki
  - Jeziki s katerimi analiziramo, testiramo in implementiramo druge programske jezike

Kaj se splača učiti

- Najbolj razširjene programske jezike.
  - C, C++, Java
  - Funkcijski | Imperativni | Objektni jeziki
- Koncepte programskih jezikov.
- Pomembne implementacijske ideje.
- Načrtovalske odločitve - tehtnice.
- Koncepti na katerih se dela raziskovalno.

Moje izkušnje

- IBM 360
  - Fortran, kartice
- Zbirnik
  - Intel 8080
- Algolova družina
  - ID80 Pascal, VAX Pascal, Turbo Pascal
  - C
- Skriptni jeziki
  - Awk, Perl
- Objektni jeziki
  - C++
  - Java
- DBMS
  - SQL, PL/SQL
- Funkcijski jeziki
  - Lisp
  - ML
  - Ocaml
- Sistemske lupine
  - csh, bash
- Logično programiranje
  - Quintus Prolog
  - Sicstus Prolog
- Web jeziki
  - JavaScript
  - ASP, PhP, JSP
- Manjši jeziki
  - Snobol