IB111 Úvod do programování skrze Python
Přednáška 13

Programovací jazyky

Nikola Beneš

14. prosinec 2016

Osnova dnešní přednášky

Programovací jazyky

- historie a vývoj
- přehled a použití

Shrnutí předmětu

- co jsme se naučili?

Co dál?

- programovací a algoritmické předměty na FI

Programovací jazyky

Kvízová otázka

Kdo byl první programátor?

Kvízová otázka

Kdo byl první programátor?

Ada, hraběnka z Lovelace.

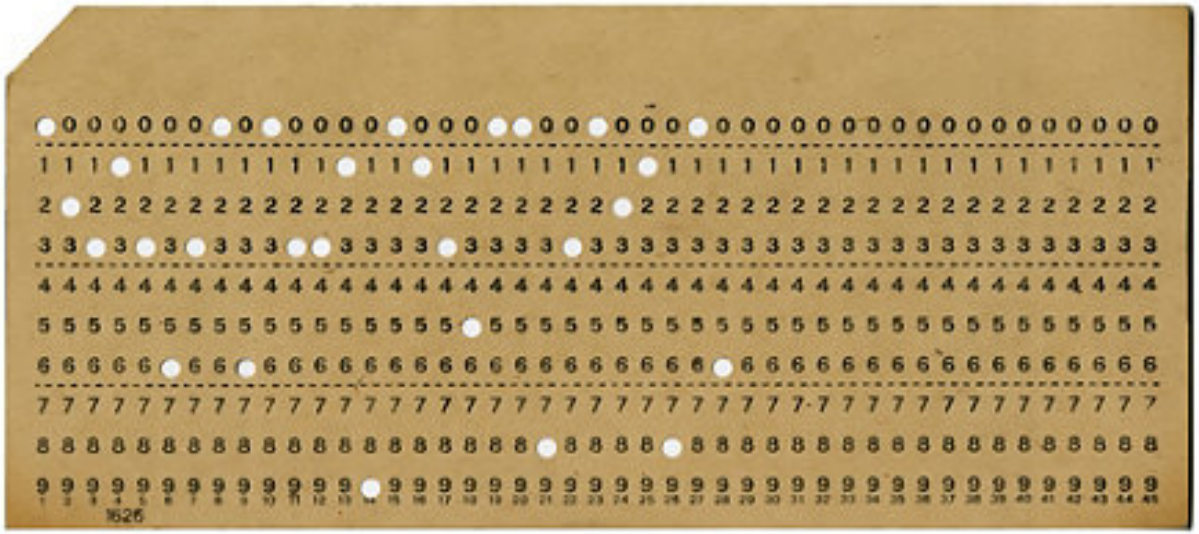
Minulý týden uplynulo 201 let od jejího narození.

(Ada Lovelace Day – 10. prosince)

Historie programovacích jazyků

Počátky

- první počítač (nikdy nesestrojen): 1837 Charles Babbage
- první programátorka: Ada, hraběnka z Lovelace
- děrné štítky
  - automatický tkalcovský stav (1801, Joseph-Marie Jaquard)
  - zpracování sčítání lidu (1890, Hermann Hollerith)



Historie programovacích jazyků

Teoretické základy programování

- lambda kalkul – Alonzo Church
- Turingův stroj – Alan Turing
- a další ...

První počítače (40. léta)

- strojový kód
- první programovací jazyky

Historie programovacích jazyků

Počátky vývoje programovacích jazyků (50. - 60. léta)

- FORTRAN
  - používaný dodnes pro vědeckotechnické výpočty
- LISP
  - některé dialekty používány dodnes, významný vliv na rozvoj funkcionálních jazyků
- ALGOL
  - významný vliv na vývoj programovacích jazyků
- a jiné (COBOL, CPL, ...)

Historie programovacích jazyků

Další vývoj (70. - 80. léta)

- strukturované programování
  - jazyk C, Ada, ...
- objektově-orientované programování
  - Simula, Smalltalk, ...
  - jazyk C++
- logické a funkcionální programování
  - Prolog, ML, Scheme, ...

Moderní programovací jazyky (90. léta a dál)

- Haskell, Python, Ruby, Java, JavaScript, C#, ...
- vývoj stále pokračuje
  - vznikají nové jazyky
  - staré jazyky se vyvíjejí a mění

Dělení programovacích jazyků

Podle míry abstrakce

- jazyky vyšší/nižší úrovně

Podle způsobu překladu a spuštění

- kompilované/interpretované

Podle paradigmatu

- imperativní/deklarativní

Další druhy

- skriptovací
- objektově-orientované
- paralelní

Většina jazyků kombinuje více přístupů.

Dělení programovacích jazyků

Podle míry abstrakce (spíše spojitý přechod)

- nejnižší programovací jazyky
  - strojový kód
  - assembler (jazyk symbolických adres)
- o něco vyšší programovací jazyky (manuální správa paměti)
  - Pascal, C, Ada, ...
  - C++
- ještě vyšší programovací jazyky (automatická správa paměti)
  - Java, Python, ...
  - moderní C++

Dělení programovacích jazyků

Podle míry abstrakce (spíše spojitý přechod)

- nejnižší programovací jazyky
  - strojový kód
  - assembler (jazyk symbolických adres)
- o něco vyšší programovací jazyky (manuální správa paměti)
  - Pascal, C, Ada, ...
  - C++
- ještě vyšší programovací jazyky (automatická správa paměti)
  - Java, Python, ...
  - moderní C++

Vývoj

- směrem k vyšším jazykům
- je vhodné pamatovat i na nižší úrovně abstrakce
  - často usnadňuje pochopení vyšších úrovní
  - abstrakce nebývají vždy dokonalé

Dělení programovacích jazyků

Podle způsobu překladu a spuštění

- kompilované jazyky (C, C++, Pascal, ...)
  - překladač (kompilátor)
  - překlad do strojového kódu
  - výhoda: rychlost za běhu
- interpretované jazyky (JavaScript, Perl, PHP, ...)
  - interpret
  - postupně čte program a vykonává příkazy
  - pomalejší, ale odpadá fáze překladu

Dělení programovacích jazyků

Podle způsobu překladu a spuštění

- kompilované jazyky (C, C++, Pascal, ...)
  - překladač (kompilátor)
  - překlad do strojového kódu
  - výhoda: rychlost za běhu
- interpretované jazyky (JavaScript, Perl, PHP, ...)
  - interpret
  - postupně čte program a vykonává příkazy
  - pomalejší, ale odpadá fáze překladu

Různé kombinace (kompromisy)

- jazyky, které mohou být překládány i interpretovány
- kompilace do mezikódu (Python, Java, ...)
  - interpretuje se mezikód
- kompilace za běhu (JIT, Just-In-Time kompilace v Javě)

Paradigmata programovacích jazyků

Imperativní jazyky

- program je posloupnost instrukcí
- instrukce jsou „rozkazy“, které určují, co má počítač dělat

Paradigmata programovacích jazyků

Imperativní jazyky

- program je posloupnost instrukcí
- instrukce jsou „rozkazy“, které určují, co má počítač dělat

Deklarativní jazyky

- program je popis toho, *co* se má udělat
- logické programování (Prolog)
  - program je popsán pomocí logických formulí
- funkcionální programování (Haskell)
  - program je popsán pomocí funkcí
  - funkce nemají vedlejší efekty
  - rekurse

Moderní jazyky kombinují více přístupů:

- funkcionální prvky v C++, Javě, Pythonu, ...

Další druhy programovacích jazyků

Skriptovací jazyky

- užitečné pro jednoduché programy, tzv. *skripty*
- usnadnění zdlouhavé manuální činnosti
- Bash, Perl, Python, ...

Další druhy programovacích jazyků

Skriptovací jazyky

- užitečné pro jednoduché programy, tzv. *skripty*
- usnadnění zdlouhavé manuální činnosti
- Bash, Perl, Python, ...

Objektově-orientované jazyky

- využívají objektového návrhu programu
- většina moderních jazyků obsahuje nějaké prvky OOP
- C++, Java, Python, Objective C, C#, ...

Další druhy programovacích jazyků

Skriptovací jazyky

- užitečné pro jednoduché programy, tzv. *skripty*
- usnadnění zdlouhavé manuální činnosti
- Bash, Perl, Python, ...

Objektově-orientované jazyky

- využívají objektového návrhu programu
- většina moderních jazyků obsahuje nějaké prvky OOP
- C++, Java, Python, Objective C, C#, ...

Paralelní výpočty

- počítání na více procesorech / více počítačích zároveň
- získává na popularitě
- podpora v moderních jazycích (např. C++11)
- proč? procesory už moc rychlejší nebudou

Jaký jazyk používat?

Programovacích jazyků je mnoho. Který si mám vybrat?

Jaký jazyk používat?

Programovacích jazyků je mnoho. Který si mám vybrat?

- špatně položená otázka
- není žádný jeden ideální jazyk použitelný pro všechny druhy úkolů
- záleží na tom, co řešíme
- může být výhodné kombinovat více jazyků/přístupů

Jaký jazyk používat?

Programovacích jazyků je mnoho. Který si mám vybrat?

- špatně položená otázka
- není žádný jeden ideální jazyk použitelný pro všechny druhy úkolů
- záleží na tom, co řešíme
- může být výhodné kombinovat více jazyků/přístupů

Doporučení

- nefixujte se na jeden jazyk
- učte se různé jazyky
- různé jazyky s sebou nesou i různé způsoby přemýšlení

Jaký jazyk používat?

Programovacích jazyků je mnoho. Který si mám vybrat?

- špatně položená otázka
- není žádný jeden ideální jazyk použitelný pro všechny druhy úkolů
- záleží na tom, co řešíme
- může být výhodné kombinovat více jazyků/přístupů

Doporučení

- nefixujte se na jeden jazyk
- učte se různé jazyky
- různé jazyky s sebou nesou i různé způsoby přemýšlení

„Kolik programovacích jazyků umíš, tolikrát jsi programátorem.“

Shrnutí předmětu

Shrnutí předmětu

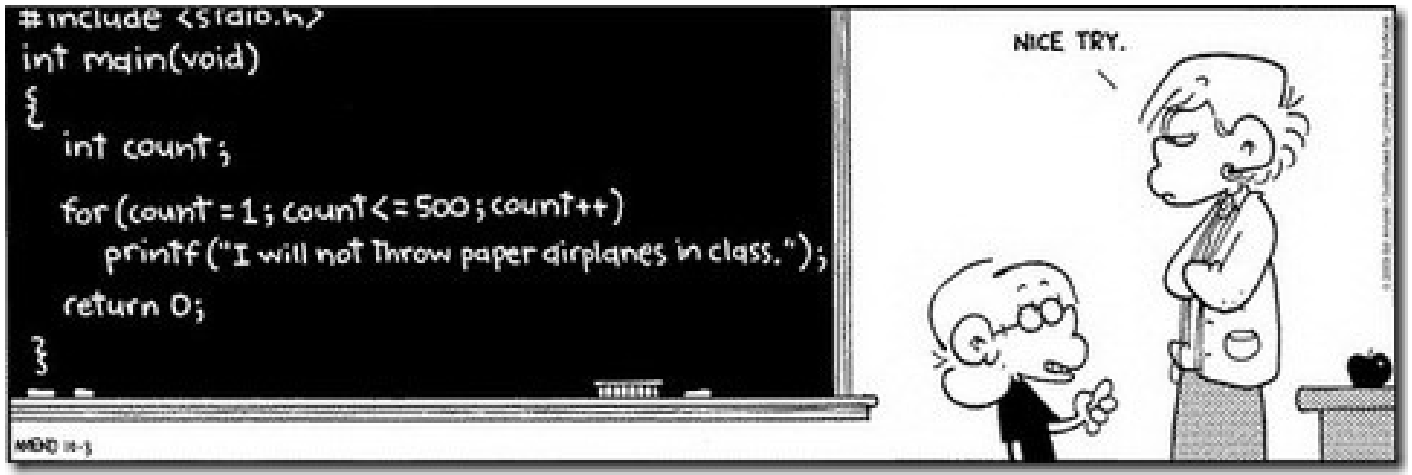
Co jsme se naučili?

- základy programování
- můžeme donutit počítač, aby dělal to, co chceme
- programování nám může usnadnit práci
- programování nám umožňuje dělat zajímavé a zábavné věci

Shrnutí předmětu

Co jsme se naučili?

- základy programování
- můžeme donutit počítač, aby dělal to, co chceme
- programování nám může **usnadnit práci**
- programování nám umožňuje dělat zajímavé a zábavné věci



Co dál?

Výhled do budoucnosti

Co dál?

- kde se dozvím víc o programování?
- jaké programovací/algoritmické předměty fakulta nabízí?
- kdo ze mě udělá lepšího programátora?

Výuka programování / algoritmizace na FI

Algoritmy (směr k větší abstrakci)

- **IB002** Algoritmy a datové struktury I
  - programátorské úlohy v Pythonu
- navazující **IV003** Algoritmy a datové struktury II

Výuka programování / algoritmizace na FI

Algoritmy (směr k větší abstrakci)

- **IB002** Algoritmy a datové struktury I
  - programátorské úlohy v Pythonu
- navazující **IV003** Algoritmy a datové struktury II

Programování podrobněji (směr k nižší abstrakci, jak funguje počítač)

- **PB071** Principy nízkoúrovňového programování
  - správa paměti, práce s řetězci, … (jazyk C)

Výuka programování / algoritmizace na FI

Algoritmy (směr k větší abstrakci)

- **IB002** Algoritmy a datové struktury I
  - programátorské úlohy v Pythonu
- navazující **IV003** Algoritmy a datové struktury II

Programování podrobněji (směr k nižší abstrakci, jak funguje počítač)

- **PB071** Principy nízkoúrovňového programování
  - správa paměti, práce s řetězci, ... (jazyk C)

Objektově-orientované programování

- **PB161** Programování v jazyce C++ (a navazující **PV264**)
- **PB162** Programování v jazyce Java (a navazující předměty)

Výuka programování / algoritmizace na FI

Algoritmy (směr k větší abstrakci)

- **IB002** Algoritmy a datové struktury I
  - programátorské úlohy v Pythonu
- navazující **IV003** Algoritmy a datové struktury II

Programování podrobněji (směr k nižší abstrakci, jak funguje počítač)

- **PB071** Principy nízkoúrovňového programování
  - správa paměti, práce s řetězci, ... (jazyk C)

Objektově-orientované programování

- **PB161** Programování v jazyce C++ (a navazující **PV264**)
- **PB162** Programování v jazyce Java (a navazující předměty)

Jiná paradigmata

- **IB015** Neimperativní programování
  - Haskell, Prolog

Výuka programování / algoritmizace na FI

Paralelní programování

- **IB109** Návrh a implementace paralelních systémů
- navazující **PV197** GPU Programming
  - používání grafických karet pro masivně paralelní výpočty

Výuka programování / algoritmizace na FI

Paralelní programování

- **IB109** Návrh a implementace paralelních systémů
- navazující **PV197** GPU Programming
  - používání grafických karet pro masivně paralelní výpočty

Více o jazyce Python a jiných jazycích

- **PV248** Kurz jazyka Python
- **PV249** Vývoj v jazyce Ruby
- **PV178** Úvod do vývoje v C#/.NET (a navazující předměty)

Výuka programování / algoritmizace na FI

Paralelní programování

- **IB109** Návrh a implementace paralelních systémů
- navazující **PV197** GPU Programming
  - používání grafických karet pro masivně paralelní výpočty

Více o jazyce Python a jiných jazycích

- **PV248** Kurz jazyka Python
- **PV249** Vývoj v jazyce Ruby
- **PV178** Úvod do vývoje v C#/.NET (a navazující předměty)

Teorie programování

- co je možné pomocí počítače řešit?
- jak rychle je to možné řešit?
- vyčíslitelnost a složitost
  - učí se v rámci různých předmětů (**IB102**, **IB107**)

A to je vše...

Dotazy, komentáře?

A to je vše...

Pěkné svátky, úspěšné zkouškové, ...

A to je vše...

Pěkné svátky, úspěšné zkouškové, ...

... a nezapomeňte programovat.