

IB111 Úvod do programování skrze Python

Úvod kurzu

2012

Dnešní přednáška

- o předmětu
- administrativa
- základy – algoritmus, programování
- motivace

Cíle předmětu

- zvládnutí základních programátorských konstrukcí
- obecné **principy** použitelné v řadě programovacích jazyků
- úvod do programátorského a algoritmického **stylu myšlení**

Úvod do programování **skrze** Python

- Python je používán pro ilustraci pojmů a příkladů, na cvičeních
- cílem není detailní zvládnutí Pythonu ani hlubší výklad specifik Pythonu
- důraz na obecné koncepty
- zvládnutí konkrétního jazyka – trénink a praxe

- máte (většina z vás) volbu mezi:
 - IB001 Úvod do programování skrze C (4+2 kr.)
 - IB111 Úvod do programování skrze Python (4+2 kr.)
 - (IB999 Vstupní test z programování (0 kr.))
- výhody a nevýhody Pythonu oproti C:
 - + příjemnější na použití
 - + snadnější začátky
 - + využití v mat. softwaru (Sage)
 - odlišná syntaxe od ostatních jazyků, které dále potkáte (C++, Java)

Co čekat: příklady

- výpočty: faktoriál, prvočísla, odmocnina, náhodná čísla
- obrázky: želví grafika, „textová grafika“, bitmapy
- jednoduché hry
- zpracování dat ze souboru, regulární výrazy

Forma předmětu

- 2h přednáška, nepovinné, ale velmi doporučené
 - slidy nemusí být pochopitelné bez komentáře
 - na cvičení často algoritmy z přednášky
 - Radek Pelánek / Zdeněk Říha
- 2h cvičení, povinné
 - programování v jazyce Python
 - více skupin
 - kapacita, prostory, ...

Hodnocení předmětu

- **závěrečná písemná zkouška:** 50 bodů
 - zkouší se principy, algoritmy, pojmy
 - „na papíře“
- **průběžná zkouška u počítače:** 20 bodů
 - programování v rámci jednoho cvičení
 - úkoly variace na příklady ze cvičení
- **domácí úkoly:** 30 bodů
 - asi 6 za semestr
- případné bonusové body
- účast na cvičení (max. 2 neomluvené hodiny)
- minimum pro ukončení: 25 bodů DÚ + průběžná zk, 25 bodů závěrečná zk

Domácí úkoly

- pracujte **samostatně**, opisování se trestá zápornými body; neřešíme, kdo opisoval
- pokud nezvládnete úlohu kompletně, zkuste alespoň něco (za méně bodů) – **jasně označte**:
 - částečné řešení
 - převzít část cizího řešení a doplnit vlastní kus
 - pozměněná (zjednodušená) úloha
- pokud řešení není úplné, uveďte v komentáři „známé nedostatky“

Relevantní agendy z ISu pro tento předmět:

- *Učební materiály* – slidy z přednášek
- *Organizační pokyny* – archiv zaslaných mailů (zadání domácích úloh)
- *Odevzdávací* – odevzdávání domácích úloh
- *Poznámkové bloky* – počet bodů z úloh

Doplňkové zdroje – knihy

- Python Programming: An Introduction to Computer Science, J. M. Zelle.
- Introduction to Computing and Programming in Python, A Multimedia Approach. M. Guzdial, B. Ericson.
- Jak to vyřešit, R. Pelánek.

Doplňkové zdroje – web

- Učíme se programovat v jazyce Python,
<http://howto.py.cz/index.htm>
- <http://interactivepython.org> – interaktivní učebnice
- Wikipedia (anglická) – články k probíraným tématům
- dokumentace k Pythonu
- Khan Academy – Computer Science

pozn. harmonogram prvních 3 týdnů ⇒ využití doplňkových zdrojů

Předpoklady

- základní počítačová gramotnost
- středoškolská matematika (např. faktoriál, prvočíslo, logaritmus)
- logické spojky (and, or, ...)

tutor.fi.muni.cz

- Binární křížovka
- Robotanik
- Matematické pexeso

Motivační úloha

- převozník, loďka uveze jen 1 další kus nákladu
- náklad: vlk, koza, zelí
- bez dozoru:
 - vlk žere kozu
 - koza žere zelí
- jak dostat vše bezpečně na druhou stranu



Motivační úloha

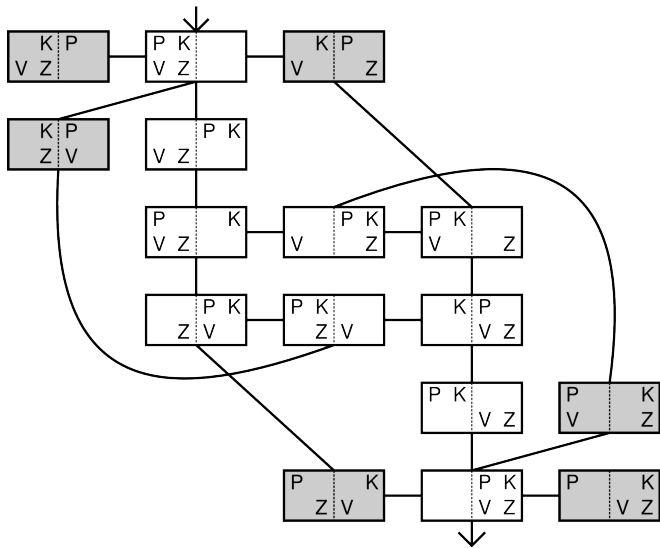
- převozník, loďka uveze jen 1 další kus nákladu
- náklad: vlk, koza, zelí
- bez dozoru:
 - vlk žere kozu
 - koza žere zelí
- jak dostat vše bezpečně na druhou stranu



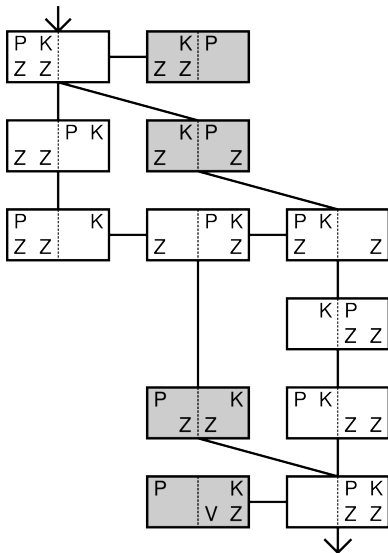
Jak řešit úlohu algoritmicky? Co to znamená?

<http://www.fi.muni.cz/~xpelanek/IB111/vkz/>

Vlk, koza a zelí



Koza a dvě zelí



- návod/postup, jak „mechanicky“ vyřešit určitý typ úlohy/problému
- typické příklady:
 - rozklad na součin prvočísel
 - nalezení nejkratší cesty mezi dvěma městy
 - vygenerovat zadání Sudoku

Žádoucí vlastnosti algoritmu

- má jasný vstup a výstup
- obecný (nejen pro omezenou třídu instancí)
- deterministický (vždy jednoznačné, jak postupovat)
- konečný, efektivní

Programování

- za **algoritmus** můžeme považovat i recept, návod
- **programování** – zápis algoritmů pro počítače
- počítače jsou „hloupé“ – zápis algoritmu musí být **opravdu přesný** (srovnej „osolíme přiměřeně“)
- nutnost vyjadřovat se přesně:
 - otrava – náročný zápis
 - bonus – nutnost myslet přesně

Programování: motivace

Proč pořádně zvládnout základy programování:

- základ pro další studium
- užitečnost
 - profesní
 - občasná
- elegance, kreativita, „síla“

Programování: způsoby využití

Zkuste vymyslet co nejrozmanitější způsoby využití programování.

Programování: způsoby využití

(příklady, rozhodně ne kompletní klasifikace)

- rozsáhlé aplikace
- programování pro web
- vestavěné systémy
- vědecké výpočty
- skriptování

*každé důraz na něco jiného, sdílí ale základní principy
„informatického myšlení“*

Samostatné rozsáhlé aplikace

- příklady:
 - kancelářský, účetní software
 - editace grafiky, zvuku, videa
 - hry
- rozsáhlé projekty
- důraz na interakci s uživatelem
- využití knihoven, práce s operačním systémem

Programování pro web

- příklady:
 - informační systémy
 - e-obchody
 - prezentace firmy
- široká škála:
 - drobné úpravy existujících systémů (CMS)
 - vytváření vlastních rozsáhlých systémů
- práce s databázemi, integrace různých prostředků (PHP, JavaScript, CSS, HTML, ...)
- důraz na soukromí – přístupová práva v IS, elektronické platby

- příklady:
 - kuchyňské spotřebiče, GPS, mobil, foťák
 - dopravní prostředky
- nízko-úrovňové programování, ovladače
- úzké propojení s konkrétním hardwarem
- bezpečnost, práce s limitovanými zdroji (paměť, energie)

- příklady:
 - simulace počasí, klimatu
 - bioinformatika (protein folding, analýza genomu, ...)
- vymýšlení algoritmů (urychlení výpočtu, distribuované výpočty)
- propojení informatiky a matematiky (příp. jiných disciplín)
- zpracování rozsáhlých dat
- uživatelské rozhraní a interaktivita jsou jen malá část

- příklady:
 - převod dat mezi různými formáty
 - rychlá analýza dat
 - prototypy, experimenty
 - drobné úpravy systému (např. správce sítě)
- malý rozsah, specifický účel
- často jednorázové aplikace

Programování v malém / ve velkém

- programování v malém
 - desítky až stovky řádků kódu
 - nezávislé na „ostatních“
 - tento předmět
- programování ve velkém
 - tisíce až milióny řádků
 - závislosti, souvislosti, návrh, testování, ...
 - další předměty (OOP, softwarové inženýrství, ...)

Programování v malém: motivace

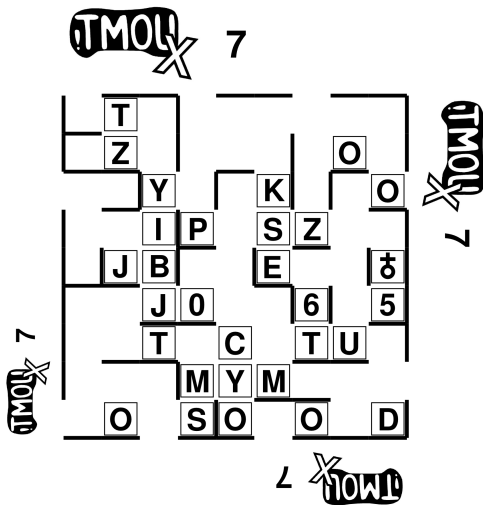
- nutná prerekvizita pro kvalitní programování ve velkém
- prakticky užitečné, i když nejste programátor na plný úvazek

Praktické programování v malém: příklady

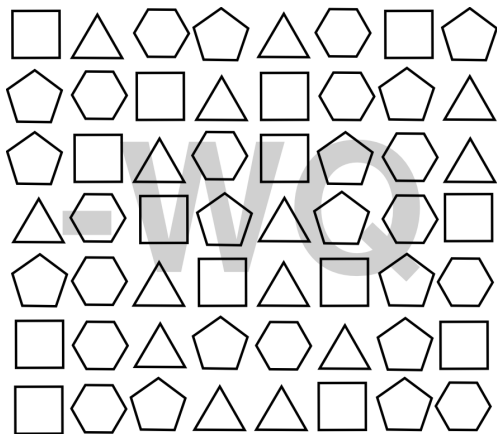
- zpracování botanických dat
- tvorba studijního katalogu
- vytvoření interaktivní webové úlohy pro systém Tutor
- generování šifer pro Tmou

pokud neovládáte X, pak vás často ani nenapadne, že by se vám X mohlo hodit

Rotující gravitační bludiště



Permutace tvarů

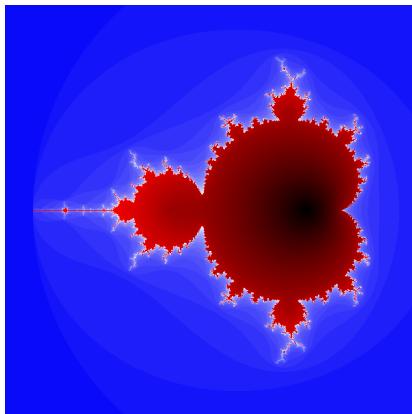


Nejen užitečnost ...

programování je zajímavé i samo o sobě

- elegantní myšlenky
- radost z objevování, experimentování
- tvoření, kreativita
- „síla“ – pár stisků klávesnice a vytvoříte něco nového a zajímavého

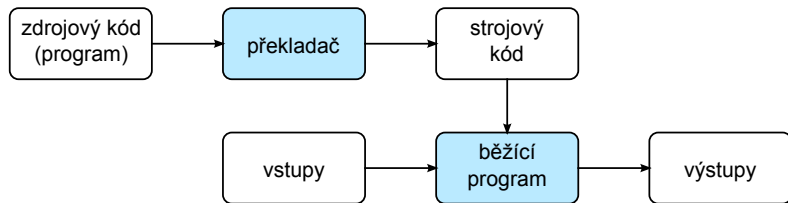
Elegance



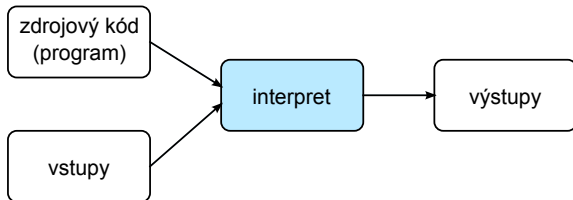
Mandelbrotova množina, 25 řádků kódu

Programovací jazyky

kompilovaný program



interpretovaný program



Programovací jazyky: klasifikace I

nízko-úrovňové

- kompilované
- nutnost řešit specifika konkrétního systému
- explicitní práce s pamětí
- náročnější vývoj (nízká efektivita práce)
- vysoká efektivita programu

vysoko-úrovňové

- interpretované
- nezávislé na konkrétním systému
- využití abstraktních datových typů
- snadnější vývoj (vysoká efektivita práce)
- nižší efektivita programu

nikoliv dvě kategorie, ale plynulý přechod; zjednodušeno

Programovací jazyky: klasifikace II

zjednodušená klasifikace a použití

nízko-úrovňové C, FORTRAN, ...

vědecké výpočty, vestavěné systémy

objektové C++, Java, C#, ...

klasické aplikace, rozsáhlé systémy

skriptovací Python, PHP, Javascript, Perl, ...

programování pro web, skriptování, prototypy

deklarativní Prolog, LISP, Haskell, ...

umělá inteligence

- **vysoko-úrovňový** – velká míra abstrakce, „spustitelný pseudokód“
- **interpretovaný** – pomalejší než kompilovaný, ale větší volnost
- **pedagogický** – byl tak navržen
- **moderní a široce používaný** – přibližně 7. nejpoužívanější jazyk
- volně a snadno **dostupný** na všech platformách

Programování v tomto kurzu

- důraz na obecné principy, nikoliv specifika Pythonu
- většina konceptů snadno a velmi podobně realizovatelná v jiných jazycích
- používáme Python 2.7

Závěrečný motivační příklad: Vězni a karty

- Albert dostane 5 karet ze standardního balíčku 52 karet
- vybere jednu z nich
- zbylé čtyři poskládá do zvoleného pořadí a dá je Bedřichovi
- Bedřich musí určit, jaká je ta pátá odstraněná karta
- Jaký systém si mají Albert s Bedřichem domluvit?