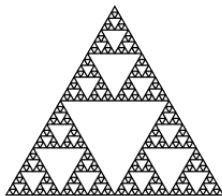
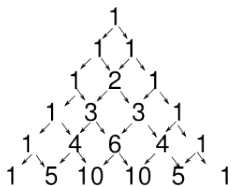
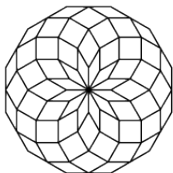


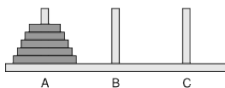
IB111 Úvod do programování skrze Python

Úvod kurzu

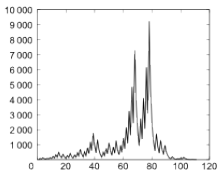
2015



P J U Y B U
 O D E O A R B A L O H E
 K A S H N U N K I K R Z
 L C Y I Z A



A -> C; A -> B; C -> B;
 A -> C; B -> A; B -> C;
 A -> C



Dnešní přednáška

- o předmětu
- organizace, ukončení
- pojmy – algoritmus, programování
- motivace, širší kontext

samotné programování až příště

Cíle předmětu

- zvládnutí základních **programátorských konstrukcí** (proměnné, funkce, if, for, while, ...)
- obecné **principy** použitelné v řadě programovacích jazyků
- úvod do programátorského a algoritmického **stylu myšlení**

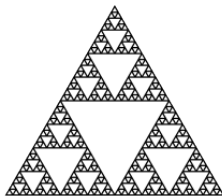
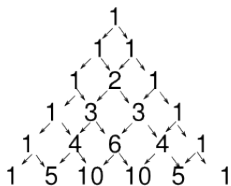
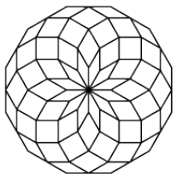
Úvod do programování **skrze** Python

- Python je používán pro ilustraci pojmů a příkladů, na cvičeních
- důraz na obecné koncepty, cílem není detailní zvládnutí Pythonu
- záměrně **ne**probíráme specifika Pythonu
- zvládnutí konkrétního jazyka – trénink a praxe

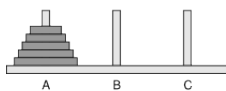
- máte (většina z vás) volbu mezi:
 - IB001 Úvod do programování skrze C (4+2 kr.)
 - IB111 Úvod do programování skrze Python (4+2 kr.)
- výhody a nevýhody Pythonu oproti C:
 - + příjemnější na použití
 - + snadnější začátky
 - + využití v mat. softwaru (Sage)
 - odlišná syntaxe od ostatních jazyků,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

Co čekat: příklady

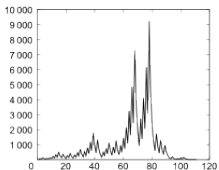
- **výpočty**: faktoriál, prvočísla, odmocnina, náhodná čísla
- **obrázky**: želví grafika, „textová grafika“, bitmapy
- **jednoduché hry**: hádání čísla, jednorozměrné piškvorky
- **zpracování dat**: statistiky dat ze souboru, regulární výrazy



P J U Y B U
 O D E O A R B A L O H E
 K A S H N U N K I K R Z
 L C Y I Z A



A -> C; A -> B; C -> B;
 A -> C; B -> A; B -> C;
 A -> C



Forma předmětu

- 2h přednáška, nepovinné, ale velmi doporučené
 - Radek Pelánek / Nikola Beneš
 - slidy nemusí být pochopitelné bez komentáře, obsahují i záměrné chyby (pro ilustraci)
 - záznamy dostupné, ale plátno nepříliš čitelné
 - na cvičení často algoritmy z přednášky
- 2h cvičení, povinné
 - programování v jazyce Python
 - více skupin, cvičících
 - speciální cvičení 01

řešení problémů, nestandardní situace, . . .

- cvičící spíše než přednášející
- osobně (po cvičení, přednášce) spíše než elektronicky
- diskuzní fórum spíše než mail

Hodnocení předmětu

- **závěrečná písemná zkouška:** 200 bodů (50 %)
 - zkouší se principy, algoritmy, pojmy
 - test s výběrem možností, podobné průběžným Odpovědníkům
- **průběžná zkouška u počítače:** 80 bodů (20 %)
 - programování v rámci jednoho cvičení
 - úkoly – variace na příklady ze cvičení
- **domácí úkoly:** 120 bodů (30 %)
 - 6 za semestr po 20 bodech
- výjimečné bonusové body

Hodnocení předmětu – minimum pro ukončení

- účast na cvičení (max. 2 neomluvené hodiny)
- odevzdání všech 6 domácích úloh (alespoň 1 bod za každou)
- alespoň 80 bodů za domácí úlohy
- alespoň 100 bodů ze závěrečné písemky
- alespoň 220 bodů celkově

Domácí úkoly

- domácí úkoly (přesné zadání, termín odevzdání) – kompetence cvičících
- 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“

Opisování

- pracujte **samostatně**
- opisování se trestá zápornými body a disciplinární komisí
- neřešíme, kdo opisoval – nesdílejte svoje řešení

Speciální cvičení

- skupina 01
- pro ty, kdo již mají programátorské zkušenosti
- bez pravidelných cvičení a domácích úloh
- 3 projekty, osobní prezentace

<http://www.fi.muni.cz/IB111/>

- harmonogram přednášek, cvičení, úkolů
- výukové materiály
- doplňující informace

Relevantní agendy z ISu pro tento předmět:

- *Učební materiály* – slidy z přednášek
- *Organizační pokyny* – archiv zaslaných mailů
- *Odpovědníky* – tréninkové testy, dobrovolné, doporučené
- *Odevzdáárny* – odevzdávání domácích úloh
- *Poznámkové bloky* – počet bodů z úloh
- *Diskuse* – nejasnosti, tipy na zajímavé zdroje, ...

Sbírka příkladů

<http://www.fi.muni.cz/IB111/sbirka/>

- interaktivní webová stránka
- příklady ze cvičení
- procvičení nad rámec cvičení
- novinka letošního běhu, postřehy \Rightarrow cvičící, diskuzní fórum IS

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.
- *Programátorská cvičebnice*, R. Pelánek.
- *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
 - dokumentace k Pythonu
 - Khan Academy – Computer Science
 - Coursera, Udacity kurzy
- např. Learn to Program: The Fundamentals, An Introduction to Interactive Programming in Python
- sdílejte užitečné zdroje v diskusním fóru předmětu

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

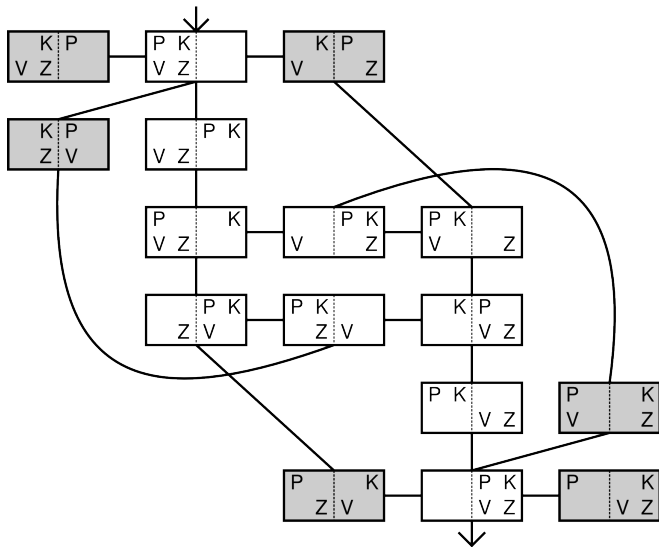
- převozník, loďka uveze jen 1 další kus nákladu
- náklad: vlk, koza, zelí
- bez dozoru:
 - vlk žere kozu
 - koza žere zelí
- jak dostat vše bezpečně na druhou stranu



Jak řešit úlohu algoritmicky? Co to znamená?

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

Vlk, koza a zelí



- návod/postup, jak „mechanicky“ vyřešit určitý typ úlohy/problému
- 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

Proč pořádně zvládnout základy programování?

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

Programování: způsoby využití

(příklady, rozhodně ne kompletní klasifikace)

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

*každé důraz na něco jiného, sdílí ale základní principy
„informatického myšlení“, námi probírané základní konstrukce
jsou potřeba všude*

- „samostatné“ aplikace pro stolní počítače, mobilní zařízení
- příklady:
 - kancelářský software
 - editace grafiky, zvuku, videa
 - hry
- 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 (Python/PHP, JavaScript, CSS, HTML, ...)
- důraz na soukromí – přístupová práva v IS, elektronické platby

- příklady:
 - kuchyňské spotřebiče, GPS, mobil, foťák
 - dopravní prostředky
 - zdravotnické přístroje
- 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 pro běžný život“

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, ...)

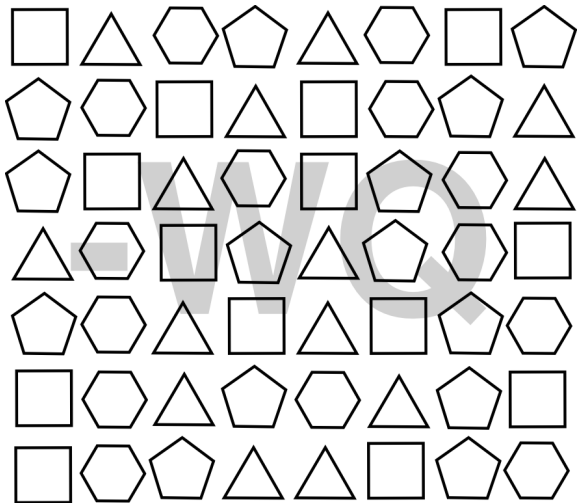
Programování v malém: motivace

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

Praktické programování v malém: příklady

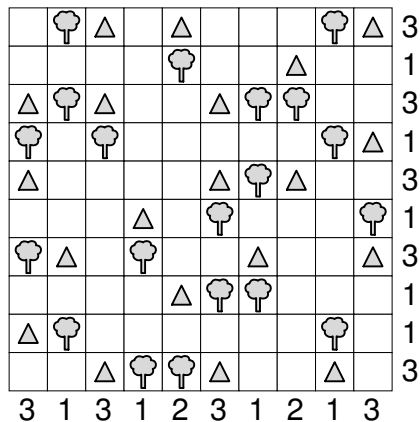
- tvorba studijního katalogu
- vytvoření interaktivní webové úlohy pro systém Tutor
- zpracování botanických dat
- vytváření šifer pro Tmou
- obrázky do knihy Hlavolamikon

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



Obrázky do knihy

"0102010902050302030703080401040304090507060606100701...;
3131313113:3131231213-;10"



Doporučené cvičení

rutinní činnost na počítači



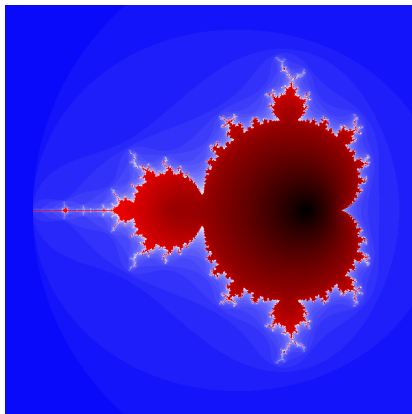
programátorské cvičení

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

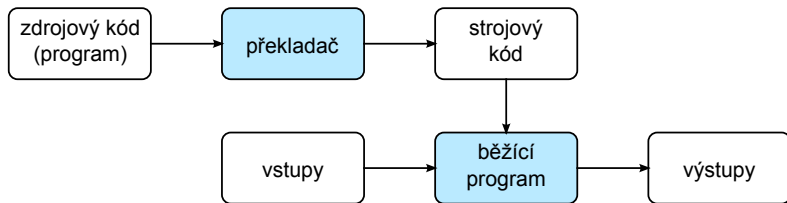
Elegance



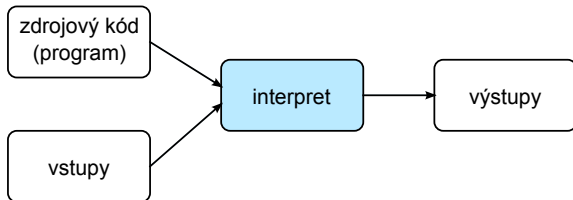
Mandelbrotova množina, 25 řádků kódu

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, ...

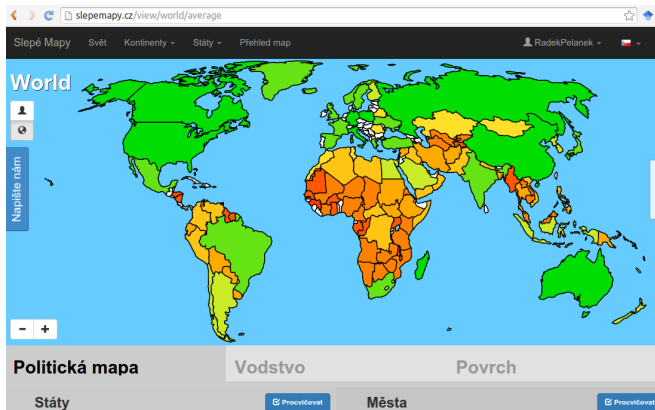
umělá inteligence

více na samostatné přednášce na konci semestru

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

Příklad aplikace: Slepé mapy

Python, Django, JavaScript, řada dílčích knihoven



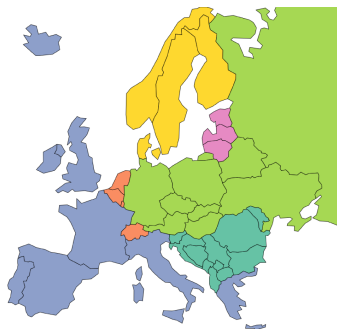
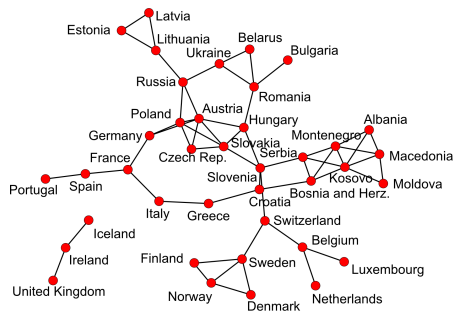
Data

```
id;user;place_asked;place_answered;type;inserted;response_time;place_map;language;options;ip_country;ip_i
10101829;124252;124;124;2;2015-05-21 07:39:18;3107;231;0;[124, 214];CZ;45280
10101830;124242;1304;1304;1;2015-05-21 07:39:19;4225;225;0;[];CZ;10877
10101831;124252;127;127;1;2015-05-21 07:39:22;3878;231;0;[73, 127];CZ;45280
10101832;123553;604;;1;2015-05-21 07:39:25;10790;126;0;[];CZ;45110
10101833;124242;1016;1017;1;2015-05-21 07:39:30;9887;225;0;[];CZ;10877
10101834;124242;1300;;1;2015-05-21 07:39:33;735;225;0;[];CZ;10877
10101835;124252;112;111;2;2015-05-21 07:39:36;13422;231;0;[54, 169, 111, 112, 214, 186];CZ;45280
10101836;124239;151;151;1;2015-05-21 07:39:39;4346;227;1;[];US;35999
10101837;123553;450;445;1;2015-05-21 07:39:40;12790;126;0;[];CZ;45110
10101838;124239;224;224;1;2015-05-21 07:39:43;2896;227;1;[];US;35999
10101839;124252;134;134;1;2015-05-21 07:39:43;4297;231;0;[134, 214, 127];CZ;45280
10101840;124239;183;183;1;2015-05-21 07:39:47;2719;227;1;[];US;35999
10101841;124239;180;180;1;2015-05-21 07:39:50;3007;227;1;[];US;35999
10101842;124252;218;207;1;2015-05-21 07:39:54;5732;231;0;[];CZ;45280
10101843;124239;87;87;1;2015-05-21 07:39:54;3145;227;1;[];US;35999
...
```

<http://www.fi.muni.cz/adaptivelearning/?a=data>

Analýza dat

Python, NumPy, SciPy, Pandas, matplotlib, Kartograph,
networkx, ...



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
- minimální použití rozšiřujících knihoven

- Základní konstrukce (proměnné, výrazy, řídicí struktury, funkce)
- Programy pracující s čísly (číselné typy, jednoduché ukázky, dělitelnost, náhoda)
- Řetězce a seznamy (a kryptografické odbočky)
- Vyhledávání a řazení (práce se seznamem, binární vyhledávání, řadicí algoritmy, základy složitosti)
- Datové typy a jejich užití (seznam, zásobník, fronta, slovník, množina)
- Rekurze (rekurzivní funkce, příklady)

- Práce se soubory, správa paměti (proměnné podrobněji)
- Složené datové typy, objekty v Pythonu
- Regulární výrazy, práce s textem, zpracování dat
- Obrázky (reprezentace, generování, úpravy)
- Vývoj programů (návrh, dekompozice problému, dokumentace, testování; případová studie)
- Programovací jazyky (přehled jazyků a jejich užití)

Hlavní návaznosti

- IB002 Algoritmy a datové struktury I
- PB161 Programování v jazyce C++
- PB162 Programování v jazyce Java
- IB015 Neimperativní programování
- PB007 Softwarové inženýrství I

Závěrečný příklad na zamyšlení: 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?