

IB113
Úvod do programování a algoritmizace
Úvod kurzu

Radek Pelánek

2022

Dnešní přednáška

- o předmětu
- organizace, ukončení
- pojmy – algoritmus, programování
- motivace, širší kontext
- představení Pythonu, rychlé demo

více o samotném programování až příště

Cíle předmětu

„dobré **základy**“

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

+ základní povědomí o pokročilých prvcích, záłudnostech, ...

Programovací jazyk Python

„Úvod do programování“

nikoliv

„Programování v Pythonu“

- 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

Programátorská kultura

Programy by měly být nejen korektní, ale i „pěkné“.

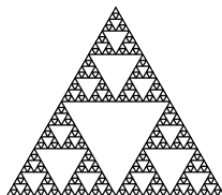
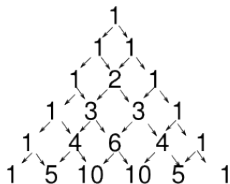
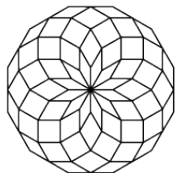
Programy by měly být nejen korektní, ale i „pěkné“.

- názvy proměnných, funkcí
- rozdělení funkcionality do funkcí
- nepoužívání „copy&paste“ kódu
- zarovnání řádku
- dokumentace, komentáře
- ...

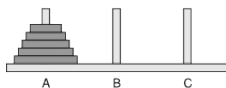
Co čekat: příklady

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

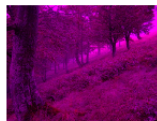
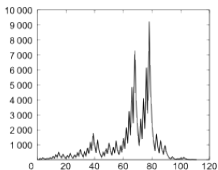
Pro textový soubor najít 20 nejčastějších slov a vypsát je v pořadí podle frekvence.



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, nepovinná, ale velmi doporučená
 - hodně „příkladový styl“
 - KVIS: opakování, kontrola porozumění, záludnosti
 - slidy nemusí být pochopitelné bez komentáře, obsahují i záměrné chyby (pro ilustraci)
 - dostupné nahrávky, ale plátno nepříliš čitelné
- 2h cvičení, povinné
 - programování v jazyce Python
 - více skupin, cvičících
- alespoň 1 hodina týdně průběžná práce

Pozn. povinnost „být připraven na cvičení“

řešení problémů, nestandardní situace, dotazy:

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

Hodnocení předmětu

500 bodů

- **domácí úkoly:** 200 bodů
 - 6 za semestr
 - bodování: 30, 30, 30, 30, 40, 40
- **vnitrosemestrální zkouška u počítače:** 100 bodů
 - programování v rámci jednoho cvičení
 - úkoly – variace na příklady ze cvičení
- **závěrečná písemná zkouška:** 200 bodů
 - zkouší se principy, algoritmy, pojmy, „čtení kódu“
 - test s výběrem možností, podobné průběžným kvízům
- výjimečné bonusové body (kladné i záporné)

Hodnocení předmětu

minimum pro ukončení:

- účast na cvičení (max. 2 neomluvené hodiny)
- odevzdání všech 6 domácích úloh (alespoň 5 bodů za každou)
- alespoň 120 bodů za domácí úlohy
- alespoň 100 bodů ze závěrečné zkoušky
- alespoň 270 bodů celkově

známka: hranice rovnoměrně mezi 270 a 500

Domácí úkoly – organizace

- skupiny: rámcově stejné zadání, odlišné detaily
- přesné zadání, termín odevzdání – kompetence cvičících

Opisování

- pracujte **samostatně**
- opisování se trestá **velmi přísně** (hodnocení F bez dalších diskuzí)
- neřešíme moc, kdo opisoval – **nesdílejte svoje řešení**

Přiznání asistence

asistence s DÚ:

- měla by být výjimečná
- **jasně přiznat** v odevzdaných materiálech
 - *celkové pojetí úlohy jsem konzultoval s XY*
 - *v této funkci jsem měl chybu a nebyl schopen jsem ji najít, použité řešení je od XY*
 - *tuto funkci jsem nezvládl napsal, použil jsem řešení ze stránky W*

<https://www.fi.muni.cz/~xpelanek/IB113/>

- harmonogram přednášek, cvičení, úkolů
- rozepsané podmínky ukončení
- výukové materiály
- doplňující informace

Studijní materiály v ISu

Relevantní agendy z ISu pro tento předmět:

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

Umíme programovat

- <https://www.umimeinformatiku.cz/>
- neomezený přístup po přihlášení, viz instrukce v mailu
- především průběžné, dobrovolné procvičování
 - konkrétní doporučení v průběhu semestru
- 2. a 4. domácí úloha (cvičení *Programování v Pythonu*)

Sbírka příkladů

<https://www.fi.muni.cz/~xpelanek/IB113/sbirka/>

- interaktivní webová stránka
- příklady ze cvičení
- procvičení nad rámec cvičení

původně určena pro paralelní předmět pro informatiky –
obsahuje i těžší příklady nad rámec IB113

Vybrané 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.

- Učíme se programovat v jazyce Python,
<http://howto.py.cz/index.htm>
- dokumentace k Pythonu
- Coursera, Udacity kurzy
- užitečné zdroje můžete sdílet v diskusním fóru předmětu

Předpoklady

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

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?

<https://www.youtube.com/watch?v=wkSMJfngxyM>

- 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

Programování: motivace

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)

Vědecké výpočty

- 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

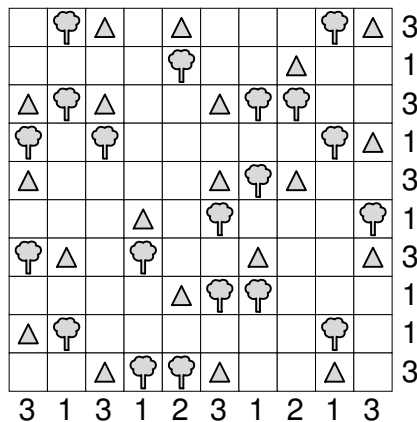
Praktické programování v malém: příklady

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

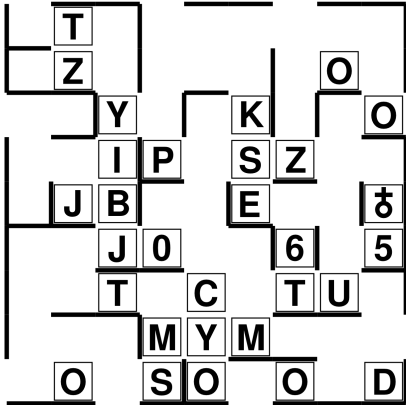
Pokud neovládáte X, pak vás často ani nenapadne, že by se vám X mohlo hodit.

Obrázky do knihy

"0102010902050302030703080401040304090507060606100701...;
313131113:3131231213-;10"



TMOX 7



TMOX 7

TMOX 7

TMOX 7

Doporučené cvičení

rutinní činnost (na počítači)



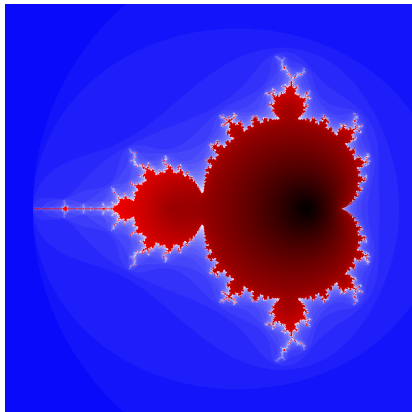
programátorské cvičení

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

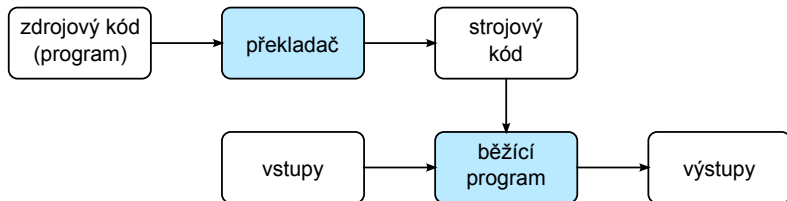


Mandelbrotova množina, 25 řádků kódu

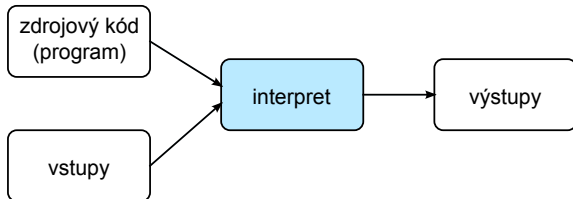
Zkuste YouTube: Mandelbrot set

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, ...

vestavěné systémy, rychlé výpočty

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** – patří mezi přibližně 5 nejpoužívanějších jazyků
- volně a snadno **dostupný** na všech platformách
- široká nabídka **knihoven**

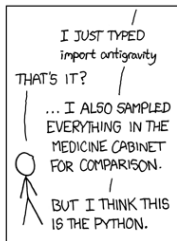
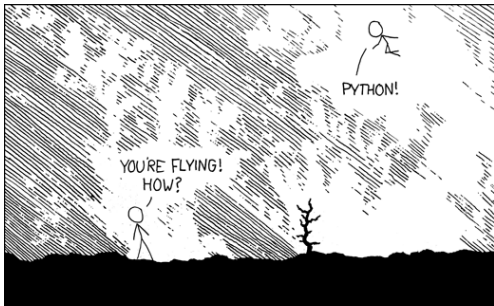
Původ jazyka a názvu

autor:
Guido van Rossum
konec 80. let

























název podle:
Monty Python's
Flying Circus





Python je používaný...

So what are the Top Ten Languages of 2018, as ranked for the typical IEEE member and *Spectrum* reader?

Language Rank	Types	Spectrum Ranking
1. Python	  	100.0
2. C++	  	99.7
3. Java	  	97.5
4. C	  	96.7
5. C#	  	89.4
6. PHP		84.9
7. R		82.9
8. JavaScript	 	82.6
9. Go	 	76.4
10. Assembly		74.1

<https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages>

Programování v tomto kurzu

- důraz na obecné principy, nikoliv specifika Pythonu
- většina konceptů snadno a velmi podobně realizovatelná v jiných jazycích
- používáme Python 3 (zpětně nekompatibilní s Python 2)
- minimální použití rozšiřujících knihoven

Osnova I: Základy

- 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 (slovník)

⇒ hlavní důraz cvičení, programátorské části zkoušky, ...

Osnova II: Pokročilejší prvky, aplikace

- Záludnosti: Proměnné, paměť, rekurze
- Práce se soubory, textem, regulární výrazy
- Složené datové typy, objekty v Pythonu
- Obrázky (reprezentace, generování, úpravy)
- Vývoj programů, programovací jazyky, praktické tipy

⇒ povědomí, základní vhled, testováno na závěrečné zkoušce

Demo základních prvků

- proměnné
- typy (číslo, řetězec, bool)
- print (funkce pro výpis)
- výrazy, operátory (aritmetické, logické)
- podmínky (if/else)
- cykly (for, while)
- bloky kódu
- funkce

- programování je
 - zajímavé
 - užitečné
- Python je
 - přívětivý i pro začátečníka
 - přímo prakticky použitelný