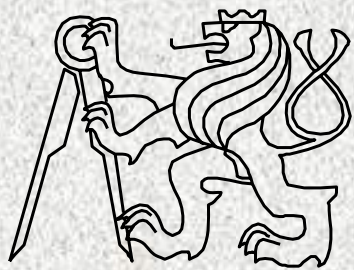
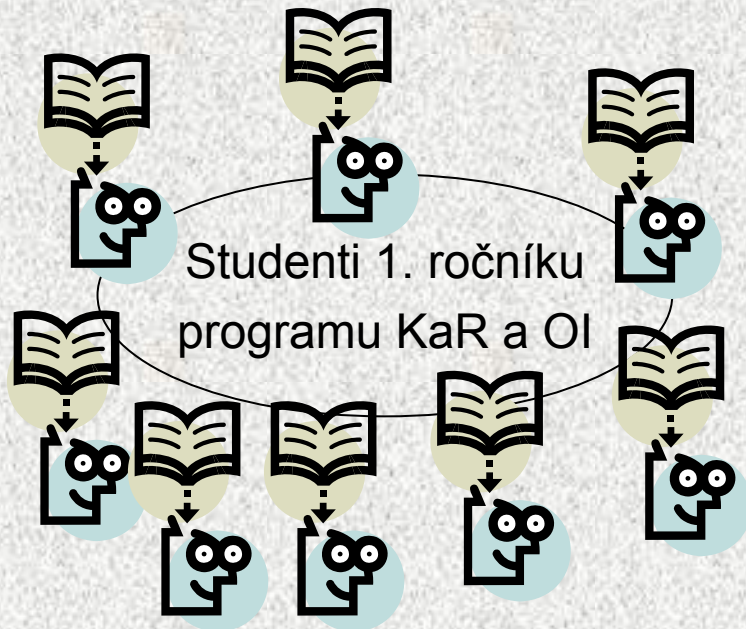


Programování 1 - úvod



A0B36PR1-Programování 1
Fakulta elektrotechnická
České vysoké učení technické

Vítejte, představme se ... ☺



Přednášející
Doc. Ing. Ivan Jelínek, CSc.

Cvičící:....



- Jste STUDENTI ČVUT FEL!
- Jste členy akademické obce!
- Děkan, senát, vědecká rada
- Zaměstnanci (tituly ...), doktorandi
- Katedry

Kdo je kdo v Programování 1

1. Vy – studenti

- Různé zkušenosti s různými programovacími jazyky

2. Přednášející:

- Doc.Ing. Jelínek Ivan CSc.

3. **Nejdůležitější učitelé: cvičící!**

- Ing. Buk Zdeněk
- Ing. R. Malinský, ved. cvičení
- Ing. J. Kuchař
- Ing. M. Balík



Proč
asi
😊??

- Garant: Doc.Ing. Jelínek Ivan CSc.

Organizace a hodnocení předmětu

- A0B36PR1 Programování 1
- Rozsah: 2p+2c
- Zakončení Z,ZK
- Kredity 6
 - po prvním semestru je nutné získat alespoň 15
 - výsledná známka ovlivní možnost tvorby rozvrhu v letním semestru

(*)Způsoby zakončení předmětu:

- zápočet
- klasifikovaný zápočet
- zápočet / zkouška

Cíl předmětu

- Programování 1

- prerekvizita Programování 2
- prerekvizita Algoritmizace

- *reprezentace dat v počítači*
- *reprezentace čísel*
- základní struktury jazyka Java
- cykly
- jednoduché programy v Javě
- ladění programů
- procedurální vs. objektový přístup
- objekty, třídy
- soubory a proudy, kolekce,
- ...

Osnova přednášek

1. **Základní pojmy** výpočetní techniky, operační systém, software, překladač, interpret, programovací jazyky, syntaxe, sémantika
2. **Koncepce Javy**, základní vlastnosti, současná podoba a vývoj, úvod do jazyka, zpracování programu, vnitřní forma, vývojová prostředí,
3. **Struktura programu**, vývoj programu, ladění programu, vývojová prostředí, proměnné, výrazy, typy, logické a číselné operátory, první program.
4. Vstup a výstup, **řídící konstrukce**, větvení, cykly
5. **Funkce**, procedury, parametry, statické proměnné, lokální proměnné, blok, princip přidělování paměti proměnným, halda, zásobník
6. **Pole**, referenční proměnná typu pole, pole jako parametr, funkce typu pole.
7. **Třídy 1** - Principy objektového přístupu, třídy, třída jako programová jednotka, třída jako zdroj funkcí, třída jako datový typ, statické a instanční metody,
8. **Třídy 2** - zapouzdření, setry, getry, metody třídy Object -equals, toString, hashCode
9. Rozklad problému na podproblémy, princip **rekurze a iterace**
10. **Spojivé struktury**, zásobník, fronta, stromy, zásobník, fronta.
11. **Úvod do ADT**, definice ADT, ADT množina, zásobník, fronta, tabulka. Implementace pomocí polí, kolekce
12. **Soubory a proudy**, soubor jako posloupnost bytů, úvod do zpracování výjimek, ukládání/čtení primitivních typů, primitivních typů a objektů (řetězců), objektů do souboru – serializace;
13. Základní principy **vyhledávání, řazení**
14. Rezerva

Osnova cvičení

1. Seznámení s počítačovou učebnou a výpočetním prostředím
2. Seznámení s vývojovým prostředím pro programování,
3. Struktura programu v jazyku Java, zadání semestrální práce
4. Sekvence, vstup, výstup, větvení
5. Cykly, odladění triviálních úloh ve vývojovém prostředí
6. Řešení složitější úlohy, rozklad na podproblémy, procedury a funkce
7. Pole
8. Třída jako datový typ
9. Třídy a dědičnost
10. Rekurze
11. Spojivé struktury a ADT
12. Test + spojivé struktury
13. Soubory a proudy
14. Zápočet

Doporučená literatura

Základní zdroje:

- Poznámky z přednášek a cvičení
- Slidy z přednášek <http://eduweb.fel.cvut.cz/courses/A0B36PR1>
- Základní příručky jazyka Java:
 - Zakhour, S: Java 6, výukový kurz, CPress, Brno, 2007
 - Herout, P.: Učebnice jazyka Java, Kopp, 2007
 - Keogh, J.:Java bez předchozích znalostí, Computer Press, 2005
 - Virius, M.: Java pro zelenáče, Neocortex, 2001

Další zdroje (publikace v češtině):

- Eckel, B.: Myslíme v jazyku Java, Grada, 2000, I + II
- Chapman, S., J.: Začínáme programovat v jazyce JAVA, Computer Press, 2001
- Pitner, T.: Java, začínáme programovat, Grada, 2002
- Hawlitzek, JAVA2, příručka programátora, Grada, 2000
- Schildt, H.: Java 2, Příručka programátora, Softpress, 2001
- Herout, P.: JAVA, grafické uživatelské prostředí a čeština, Kopp, 2001

Hodnocení a zkouška

Zdroje bodů pro hodnocení	Body
aktivita a DÚ na cvičeních	20 b (min. 10 b)
semestrální práce	30 b (min. 20 b)
test u počítače na cvičeních	20 b (min. 10 b)
písemný zkouškový test	20 b (min. 10 b)
Ústní zkouška	20 b (-10b pokud k ústní)

Minimální počet bodů pro zápočet je 40 bodů

Body ze cvičení, maximálně 70. 60 a více bodů → možnost A, B, C podle prémiového testu

Klasifikace na základě bodového hodnocení)				
klasifikace	počet bodů	číselně	slovně	
A	90 - 100	1	výborně	
B	80 - 89	1,5	velmi dobře	
C	70 - 79	2	dobře	
D	60 - 69	2,5	uspokojivě	
E	50 - 59	3	dostatečně	
F	< 50	4	nedostatečně	

Možnost nechat si zapsat známku nebo jít k ústní zkoušce – odečte se 10 bodů

Zakončení předmětu: zápočet, **zkouška** (na základě bodového hodnocení)

Začínáme



Šest zákonů programování

1. V každém programu je alespoň jedna chyba
2. Každý program lze zkrátit alespoň o jeden řádek
3. Nejjednodušší chyby se nejhůře hledají
4. Každou opravou se do programu zanesou nová chyba
5. Když už se zdá, že program je v pořádku, určitě jste něco přehlédli
6. Programátor dělá to co umí, počítač si dělá, co chce

Začínáme doopravdy



Algoritmus

- Algoritmus
 - postup při řešení určité třídy úloh, který je tvořen seznamem **jednoznačně definovaných příkazů** a zaručuje, že pro **každou přípustnou kombinaci vstupních dat** se po provedení **konečného počtu kroků** dospěje k **požadovaným výsledkům**
- Vlastnosti algoritmu:
 - ***hromadnost***
měnitelná vstupní data
 - ***determinovanost***
každý krok je jednoznačně definován
 - ***konečnost a resultativnost***
pro přípustná vstupní data se po provedení konečného počtu kroků dojde k požadovaným výsledkům
- Algoritmus – syntetický model postupu řešení obecných úloh
- Prostředky pro zápis algoritmu
 - přirozený jazyk, vývojové diagramy, struktogramy, pseudojazyk, programovací jazyk

Algoritmus, slovní popis, analýza

- Úloha:
Najděte největšího společného dělitele čísel 6 a 15
- Řešení:
Popište postup tak, aby byl použitelný pro dvě libovolná přirozená čísla, nejen pro 6 a 15:
 - označme zadaná čísla x a y a menší z nich d
 - není-li d společným dělitelem x a y , pak zmenšíme d o 1, test opakujeme a skončíme, až d bude společným dělitelem x a y
- Poznámka:
Význam symbolů x , y a d použitých v algoritmu:
 - jsou to **proměnné** (paměťová místa), ve kterých je uložena nějaká hodnota, která se může v průběhu výpočtu měnit

Algoritmus – „průběh výpočtu“

Úloha: najděte největšího společného dělitele čísel 6 a 15

Průběh řešení:

krok	x	y	d	poznámka
	6	15	?	zadání vstupních dat
1	6	15	6	
2	6	15	6	d není dělitelem y , proved' krok zmenšení d
3	6	15	5	
2	6	15	5	d není dělitelem x , proved' krok zmenšení d
3	6	15	4	
2	6	15	4	d není dělitelem x ani y , proved' krok zmenšení d
3	6	15	3	
2	6	15	3	d je dělitelem x i y , proved' krok 4
4	6	15	3	výsledek je hodnota 3

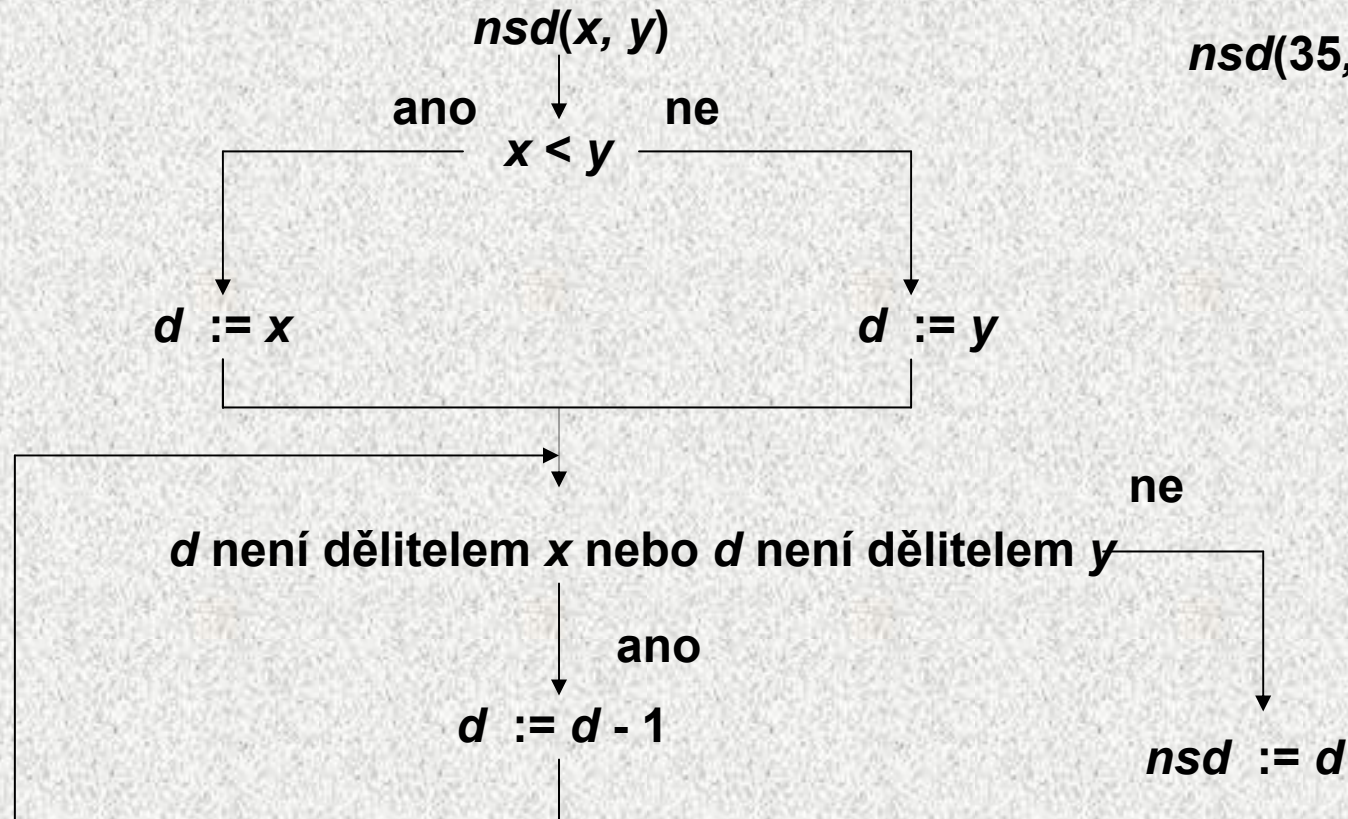
Algoritmus - „zápis“

- Úloha:
najděte největšího společného dělitele
- Obecný popis:
Vstup: přirozená čísla x a y
Výstup: $nsd(x,y)$
Postup:
 1. Je-li $x < y$, pak d má hodnotu x , jinak d má hodnotu y
 2. Opakuj krok 3, pokud d není dělitelem x nebo d není dělitelem y
 3. Zmenši d o 1
 4. Výsledkem je hodnota d
- Sestavili jsme algoritmus pro výpočet největšího společného dělitele dvou přirozených čísel

Algoritmy, vývojový diagram

- Vývojový diagram

Výpočet pro
 $nsd(35, 7)$



Programy a programovací jazyky

- Program je předpis (**zápis algoritmu**) pro provedení určitých akcí počítačem zapsaný v programovacím jazyku
- Programovací jazyky
 - **strojově orientované**
 - strojový jazyk = jazyk fyzického procesoru
 - assembler (jazyk symbolických adres)
 - **vyšší jazyky**
 - **imperativní** (příkazové, procedurální)
 - neimperativní (např. funkcionální)
- Hlavní rysy imperativních jazyků (např. C, C++, **Java**, Pascal, Basic, ...)
 - zpracovávané údaje mají formu datových objektů různých typů, které jsou v programu reprezentovány pomocí proměnných resp. konstant
 - program obsahuje deklarace a příkazy
 - deklarace definují význam jmen (identifikátorů)
 - příkazy předepisují akce s datovými objekty nebo způsob řízení výpočtu

Algoritmy a programy

- Zápis algoritmu v pseudojazyku

```
nsd(x,y):
```

```
  if x<y then d:=x else d:=y;
```

```
  while d „není dělitelem“ x or d „není dělitelem“ y do
```

```
    d:=d-1;
```

```
  nsd:=d;
```

int .. celé číslo,
proměnné x,y,d a
výsledek budou typu
int

- Zápis algoritmu v programovacím jazyku

```
int nsd(int x, int y) {
```

```
  int d;
```

```
  if (x<y) d=x; else d=y;
```

```
  while (x%d!=0 || y%d!=0) d--;
```

```
  return d;
```

```
}
```

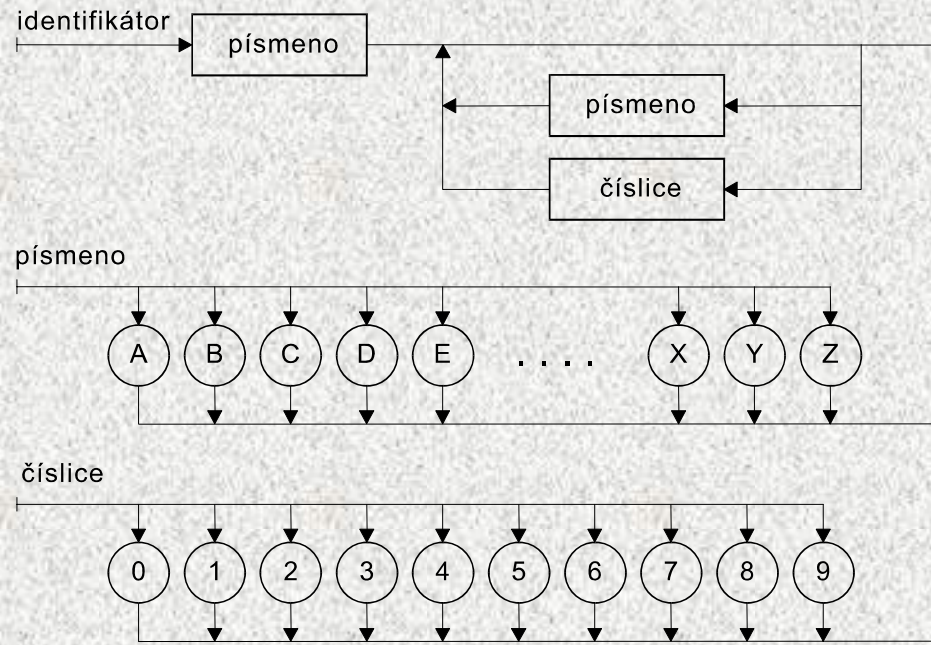
d--; příkaz, který sníží
hodnotu uloženou v
proměnné d o jedničku

x%d ... zbytek po dělení čísla x číslem
d,
!= ... není rovno,
|| ... nebo

Vlastnosti programovacích jazyků

- **Syntaxe**

- souhrn pravidel udávajících přípustné tvary dílčích konstrukcí a celého programu
- syntaktické diagramy



- **Sémantika**

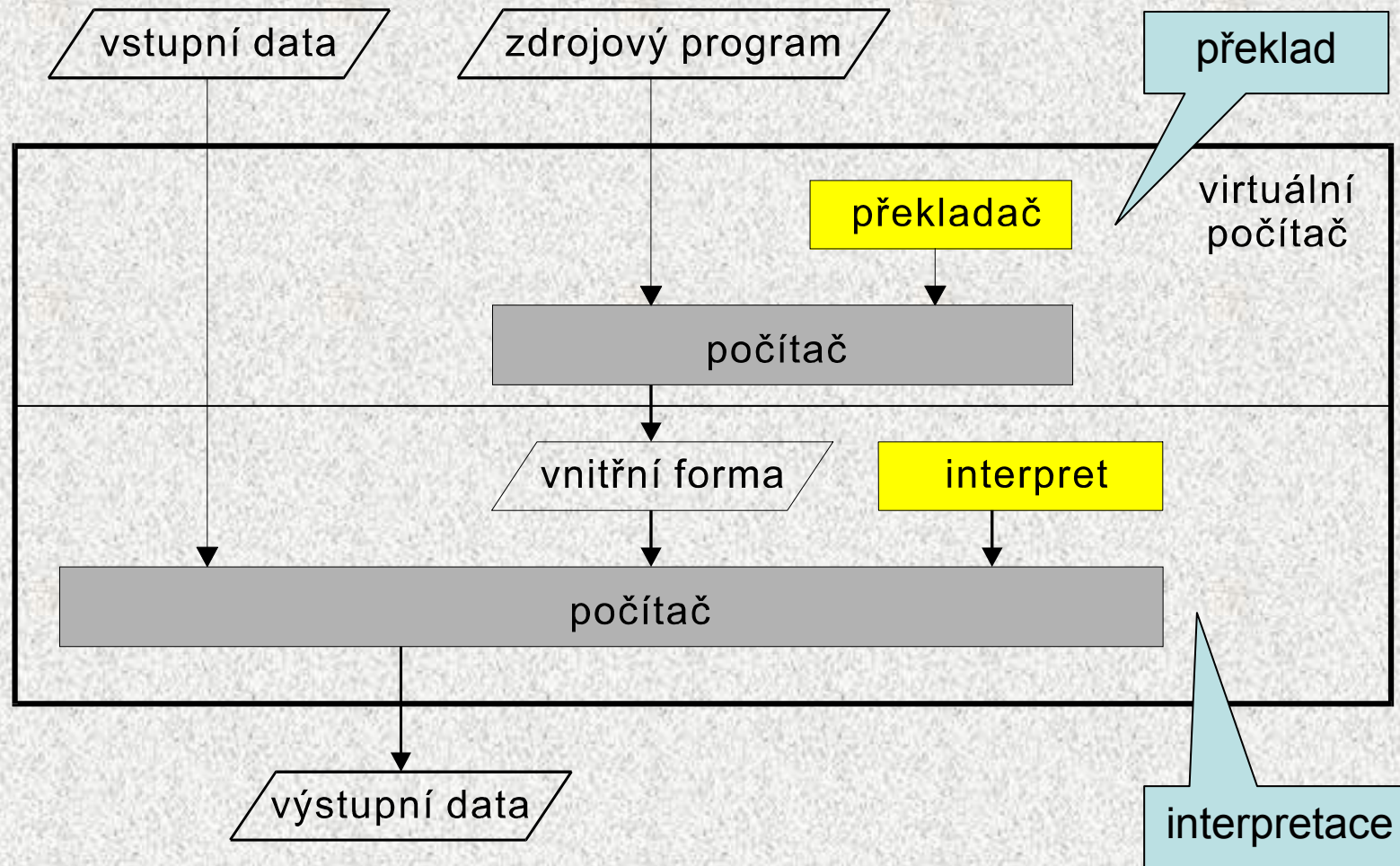
- udává význam jednotlivých konstrukcí

Rozšířená BNF

- Rozšířená Backus-Naurova forma – EBNF
- Příklad: identifikátor
 - identifikátor = písmeno {písmeno | číslice}
 - písmeno = 'A' | 'B' | 'C' | 'D' | ... | 'X' | 'Y' | 'Z'
 - číslice = '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'
- Neterminály:
 - identifikátor, písmeno, číslice
- Terminály:
 - 'A', 'B', ...
- Význam metasymbolů:
 - {x} žádný nebo několik výskytů x
 - x | y x nebo y
 - [x] žádný nebo jeden výskyt x

Implementace programovacích jazyků

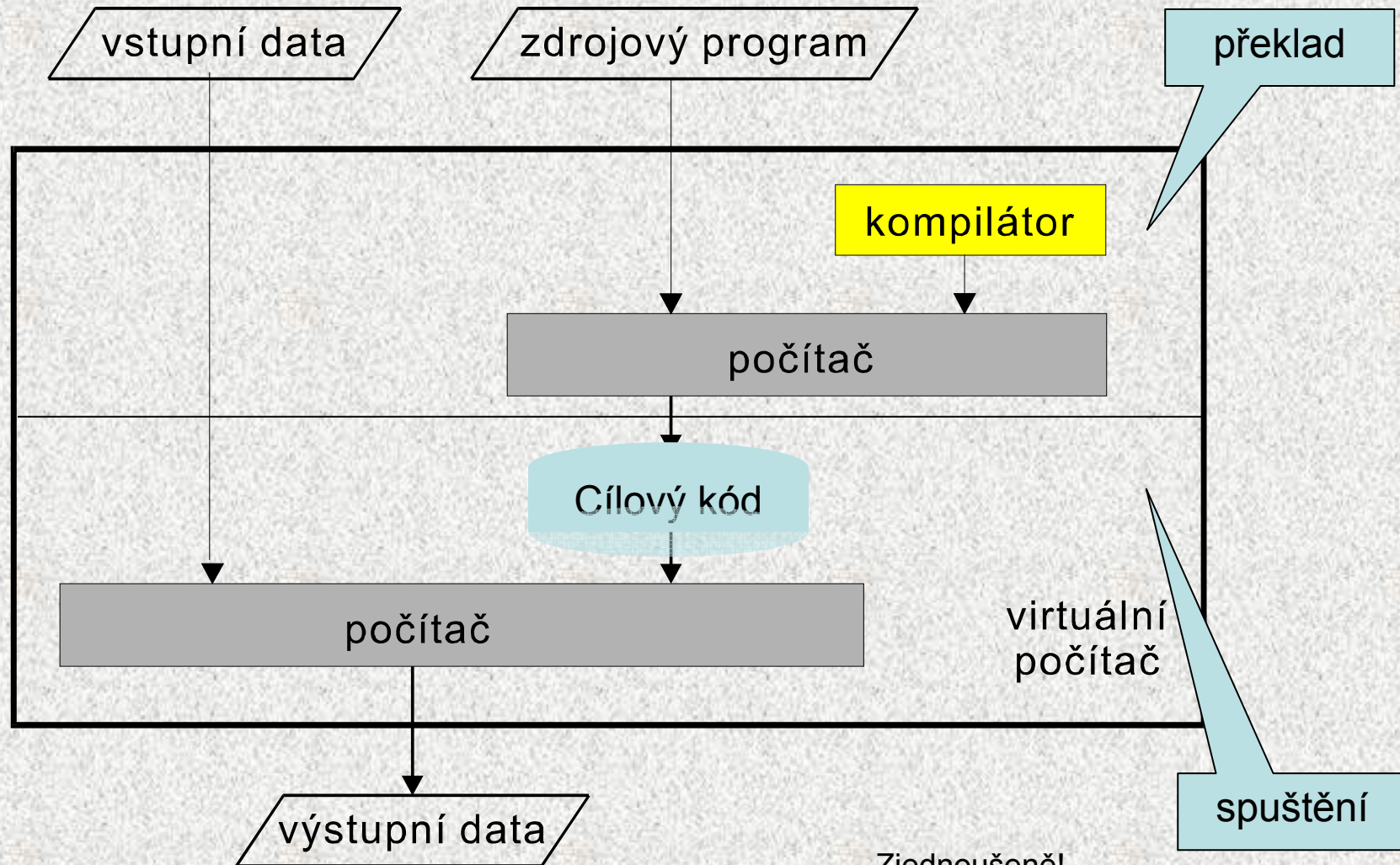
- Interpretační metoda:



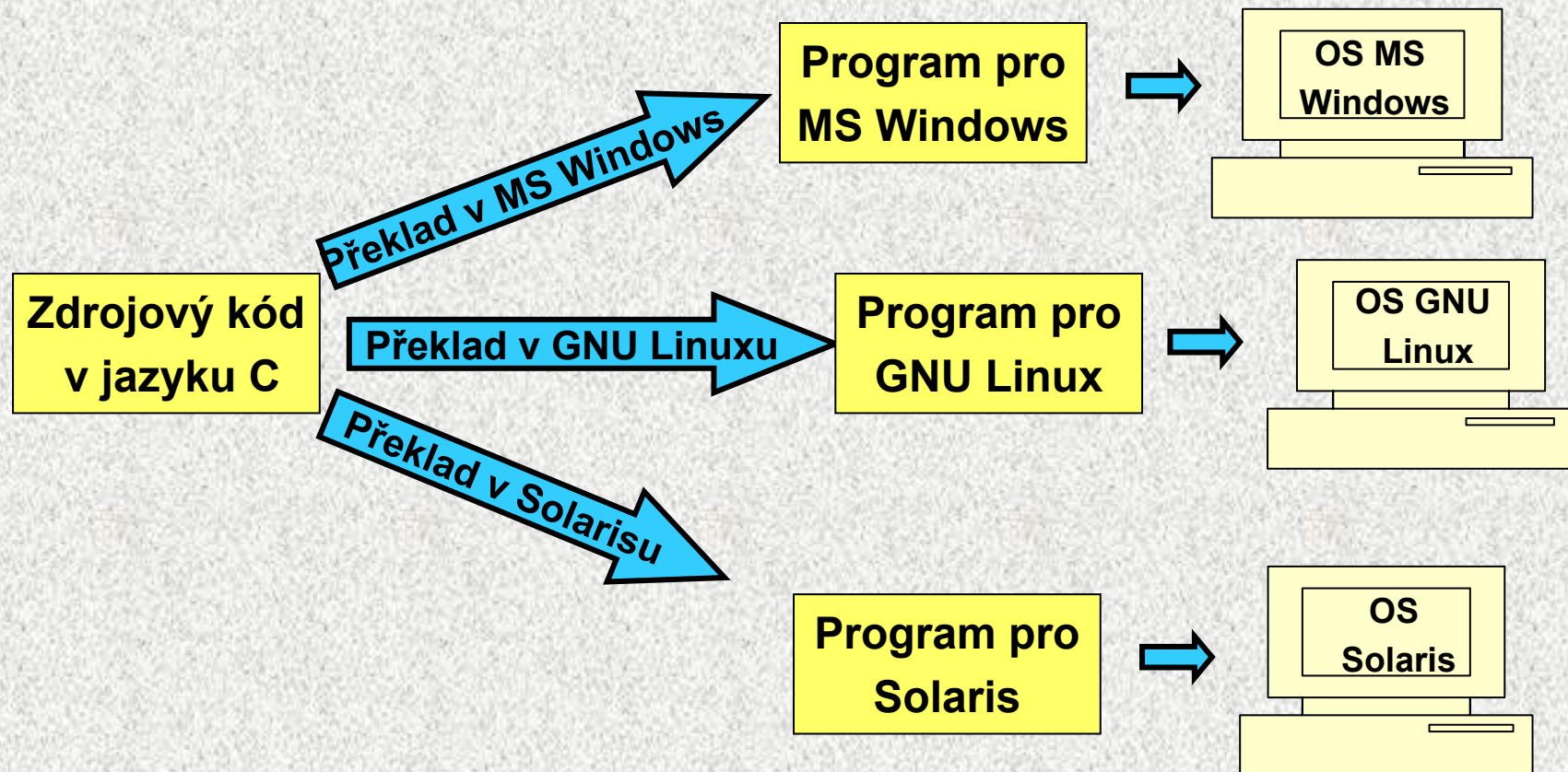
Zjednoušeně!

Implementace programovacích jazyků

- Kompilační metoda:



Kompilační metoda - jazyk C, C++

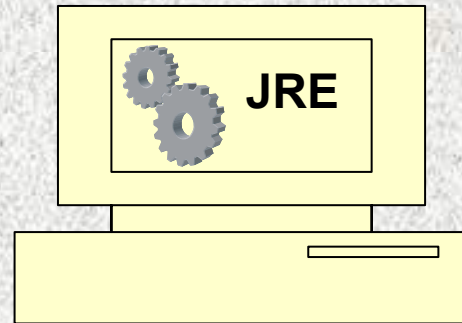
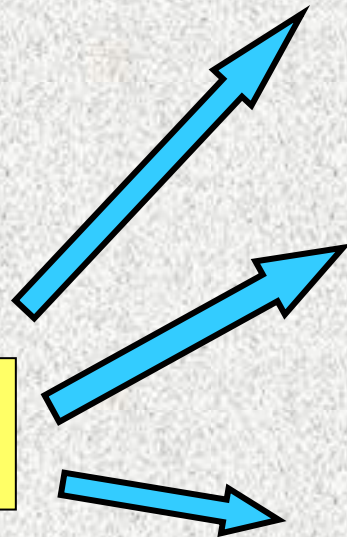


Interpretační metoda - jazyk Java

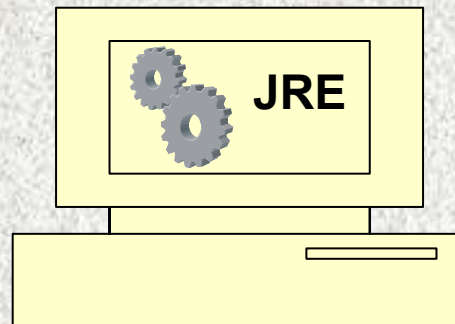
Zdrojový kód
v jazyku Java
soubor `.java`



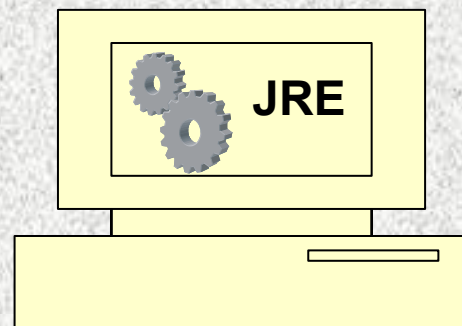
Bytecode
soubor `.class`



OS MS
Windows

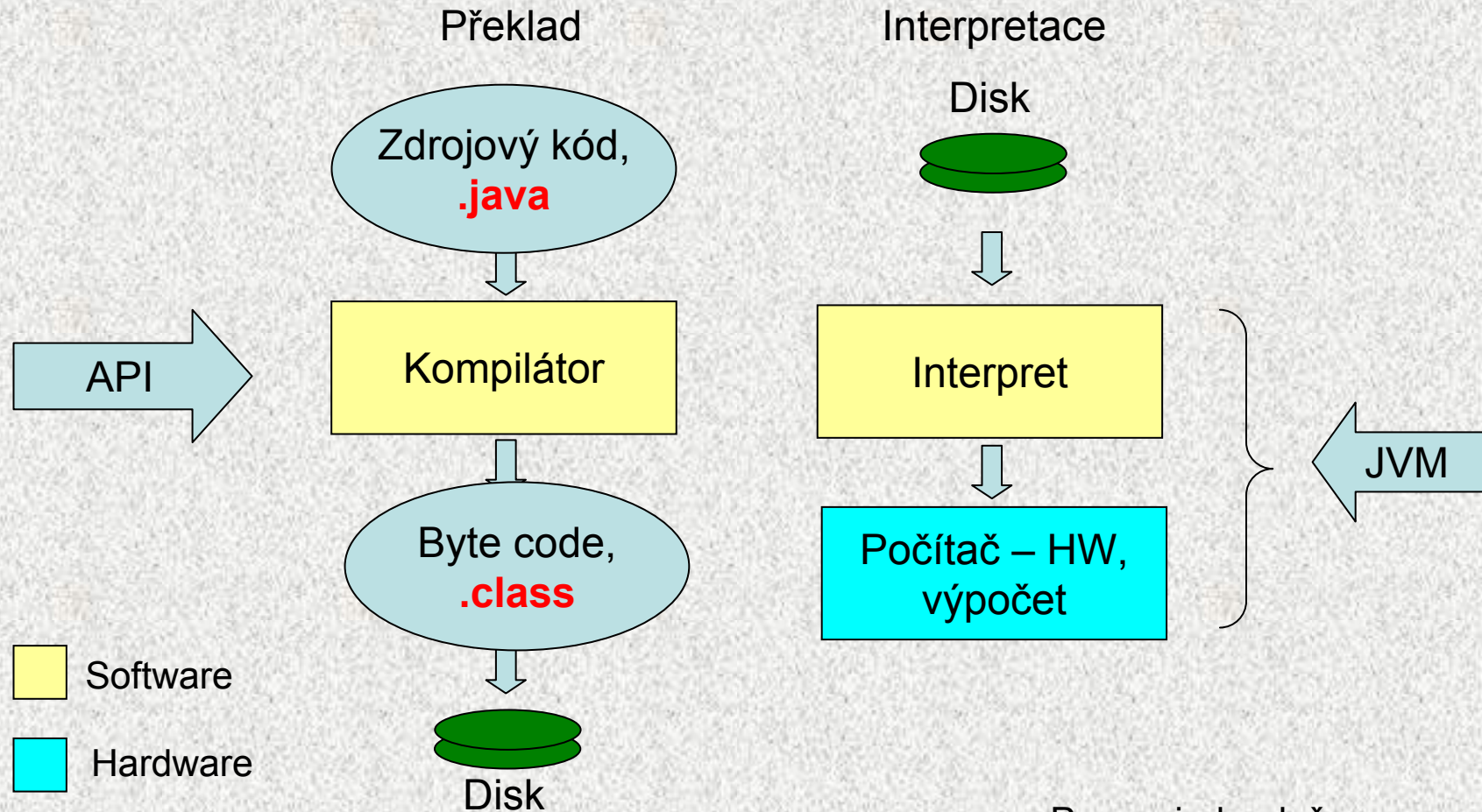


OS GNU
Linux



OS
Solaris

Java Platforma (JRE) = Java Core API + JVM

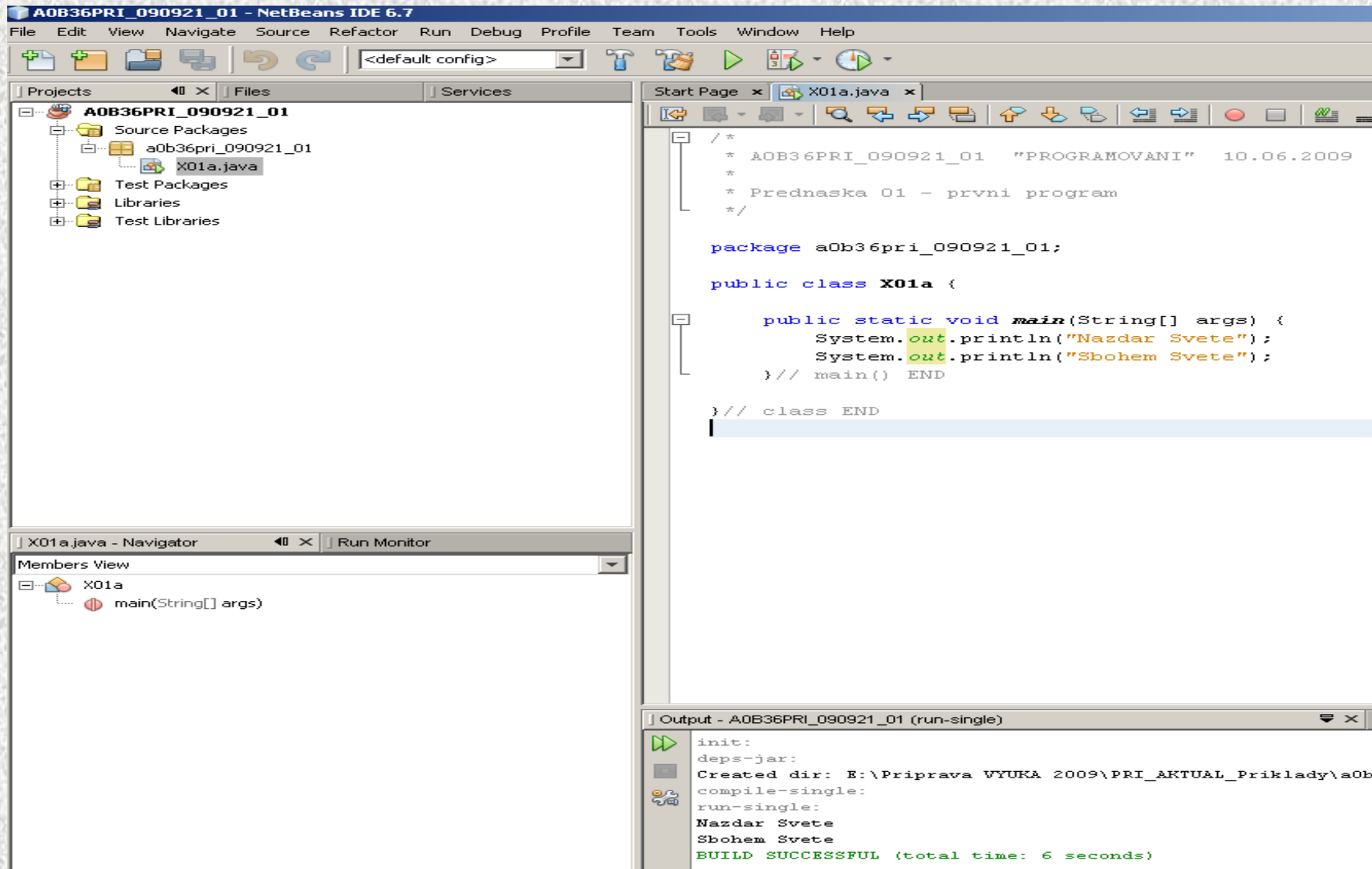


Pozn: zjednodušeno

IDE – vývojový nástroj

- Pro vývoj programů se používá vývojový nástroj nazývaný IDE (Integrated Development Environment)
- IDE je v dnešní době k dispozici pro všechny běžné používané programovací jazyky, často od různých výrobců software
- V předmětu A0B36PR1 budeme pro vývoj programů v Javě používat IDE NetBeans
- IDE NetBeans je možné zdarma stáhnout na webu a nainstalovat na platformě Windows nebo Unix (viz <http://www.netbeans.org>)
- NetBeans obsahují (nebo využívají) všechny nezbytné části pro vývoj programu v Javě (textový editor, kompilátor, ladící prostředky a další části)
- S používáním NetBeans se seznámíte na cvičeních

NetBeans



Jazyk JAVA - úvod

- Jazyk Java je implementován **interpretačním** způsobem
 - program je tvořen jedním nebo několika **zdrojovými soubory** s příponou **.java**:

`Program.java`

- zdrojové soubory se přeloží **překladačem(*) javac** do **vnitřní formy** (byte code, bajt-kód) s příponou **.class**:

`Program.java > javac > Program.class`

- **interpretaci** vnitřní formy **provede program java** (JVM – Java Virtual Machine v balíčku **JRE** Java Runtime Environment) a provede výpočet:

`Program.class > java > „výpočet“`

Poznámky:

(*) v terminologii firmy Sun to je kompilátor

- program obvykle **využívá řadu knihoven (Java Core API)**, které je třeba mít k dispozici jak při překladu, tak při interpretaci!!!

JAVA – první program

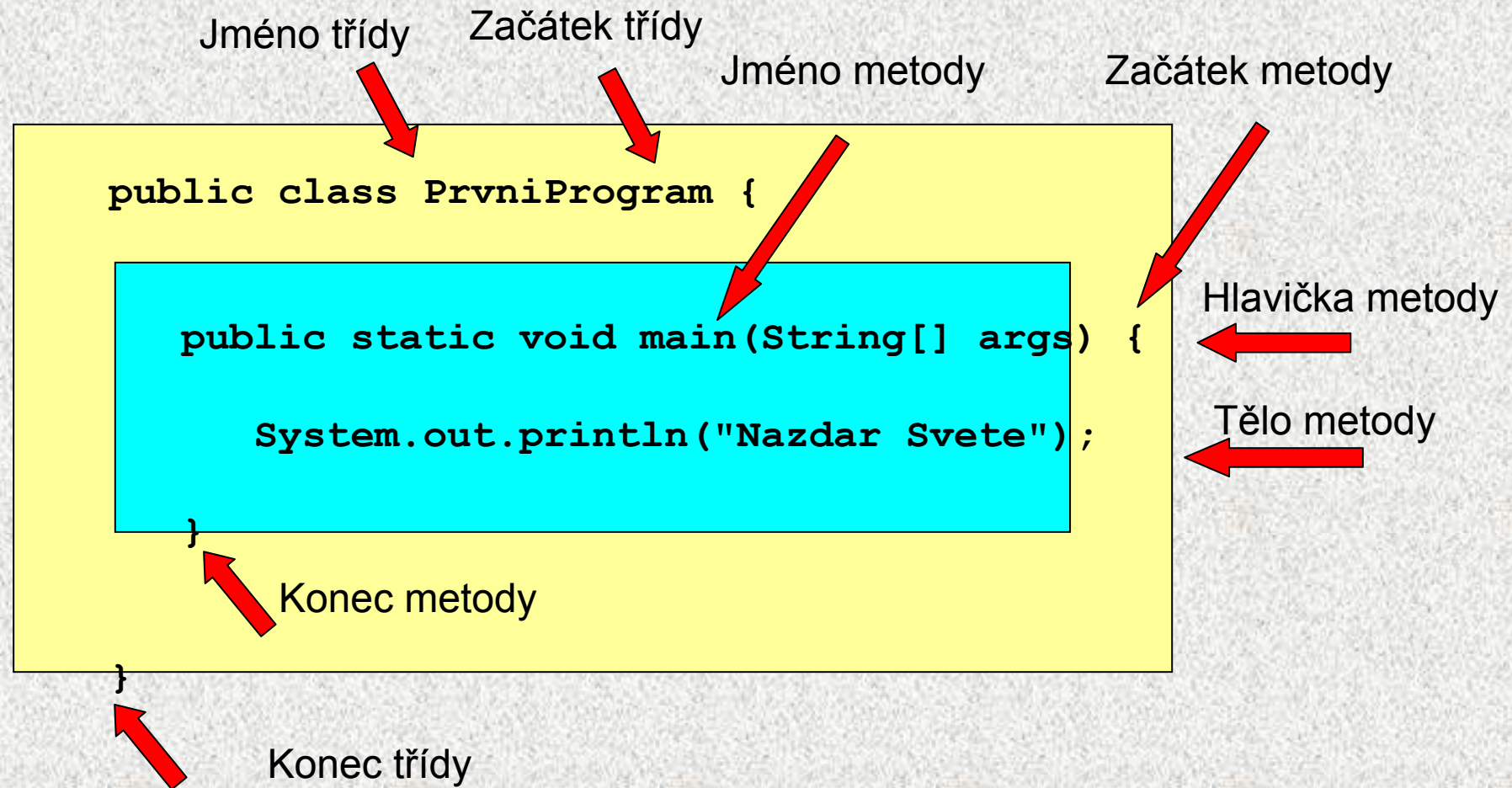
- Příklad: program vypíše daný text na obrazovku:

```
public class PrvniProgram {  
    public static void main(String[] args) {  
        System.out.println("Nazdar Svete");  
    }  
}
```

- Po překladu a spuštění se na obrazovku vypíše
Nazdar Svete
- Nejjednodušší zdrojový program – je uložen v jediném souboru. Jméno souboru musí být shodné se jménem třídy (zde PrvniProgram) a přípona (rozšíření) jména souboru je povinná .java (naš program bude tedy uložen v souboru „PrvniProgram.java“)
 - **deklarace veřejné třídy (public class),**
 - **hlavní funkce main (veřejná statická metoda, public static method)**
- Hlavička funkce funkce main ():
 - klíčová slova **public static void** (void - procedura)
 - **(String[] args)** specifikace vstupních parametrů
- Konvence: jména tříd se píší s prvním velkým písmenem

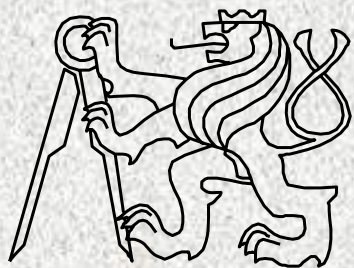
JAVA – bloková struktura

- Program má blokový charakter (blok třídy, blok(y) metod(y))
 - Nejtriviálnější program je tvořen metodou `main` ve třídě (třída = program)



Programování 1 – úvod

KONEC



A0B36PR1-Programování 1
Fakulta elektrotechnická
České vysoké učení technické

Vlastní studium



Základní pojmy informatiky

- *Definujeme základní pojmy, shrneme důležité vlastnosti a vztahy (u řady z nich se předpokládá znalost!!)*
 - *Informatika*
 - *Software*
 - *Hardware*
 - *Operační systém*
 - *Algoritmus*
 - *Programovací jazyk*
 - *Překladač*
 - *Kompilátor*
 - *Reprezentace dat v počítači*

Informatika

Computer science is no more about computers than astronomy is about telescopes.

Informatika se nezabývá počítači o nic více než astronomie dalekohledy.

- Edsger Dijkstra, informatik

- zabývá se zpracováním informací nejen na počítačích
- studuje výpočetní a informační procesy z hlediska hardware i software – „*technická* informatika“
- je součástí teorie informací, věda spojující aplikovanou matematiku a elektrotechniku za účelem kvantitativního vyjádření informace

Software – programové vybavení

- V informatice sada všech počítačových programů v počítači
- Software zahrnuje
 - operační systém (zajišťuje běh programů)
 - aplikační software (pracuje s ním uživatel)
 - další (knihovny, middleware, BIOS, firmware apod.)


Hardware

Zahrnuje všechny fyzické součásti počítače

- čistě elektronická zařízení (procesor, paměť, display)
- elektromechanické díly (klávesnice, tiskárna, diskety, disky, jednotky CD-ROM, páskové jednotky, reproduktory) pro vstup, výstup a ukládání dat.
- Počítač se skládá z procesoru, operační paměti a vstupně-výstupních zařízení.

Operační systémy - dělení

- OS mainframů (velké počítače pro kritické aplikace) - VMS, OS/360,...
- OS osobních počítačů řady PC - HP-UX, Solaris, BSD, MS DOS, MS Windows,...
- OS osobních počítačů řady Apple - Systém 1..Systém 9, Mac OS X,...
- OS kapesních počítačů, PDA, komunikátorů a smartphonů - Symbian, PalmOS, Android,...
- OS pro vestavné (embedded) systémy
- Smart Card OS
- Reálné časové OS



Kterých
je nejvíce??

Reprezentace dat v počítači

```
00000000: CA FE BA BE 00 00 00 31|00 19 0A 00 05 00 15 07 | Eţşl'■■■1■■■■■■■■■
00000010: 00 16 0A 00 02 00 15 07|00 17 07 00 18 01 00 06 | ■■■■■■■■■■■■■■■■■■
00000020: 3C 69 6E 69 74 3E 01 00|03 28 29 56 01 00 04 43 | <init>■■■()V■■■C
00000030: 6F 64 65 01 00 0F 4C 69|6E 65 4E 75 6D 62 65 72 | ode■■■LineNumber
00000040: 54 61 62 6C 65 01 00 12|4C 6F 63 61 6C 56 61 72 | Table■■■LocalVar
00000050: 69 61 62 6C 65 54 61 62|6C 65 01 00 04 74 68 69 | iableTable■■■thi
00000060: 73 01 00 1A 4C 64 61 74|61 62 61 7A 65 72 69 64 | s■■■Ldatabazetid
00000070: 69 63 75 32 2F 53 74 61|72 74 47 55 49 3B 01 00 | icu2/StartGUI;■■
00000080: 04 6D 61 69 6E 01 00 16|28 5B 4C 6A 61 76 61 2F | ■main■■■([Ljava/
00000090: 6C 61 6E 67 2F 53 74 72|69 6E 67 3B 29 56 01 00 | lang/String;)V■■
000000A0: 04 61 72 67 73 01 00 13|5B 4C 6A 61 76 61 2F 6C | ■args■■■[Ljava/l
000000B0: 61 6E 67 2F 53 74 72 69|6E 67 3B 01 00 03 67 75 | ang/String;■■■gu
000000C0: 69 01 00 15 4C 64 61 74|61 62 61 7A 65 72 69 64 | i■■■Ldatabazetid
```

Elektronická paměť

Součástka, zařízení nebo materiál, který umožní uložit obsah informace (zápis do paměti), uchovat ji po požadovanou dobu a znovu ji získat pro další použití (čtení paměti)

- Vnitřní paměť (primární)
- Vnější paměť

Vnitřní paměť

- **Registry procesoru** - několik (až desítky) registrů, ukládání operandů a výsledků aritmetických a logických operací, nejrychlejší paměť připojená k procesoru (stejně rychlá, jako procesor)
- **Cache** - pro urychlení komunikace s pamětí, rychlá statická paměť, u novějších procesorů velikost stovky kB až MB, více úrovní, přičemž číslo určuje vzdálenost od procesoru
 - L1 – typicky přímo na procesoru
 - L2 – například na destičce s procesorem (tzv. boxované procesory)
 - L3 – na základní desce
- **Operační paměť RAM** - pomalejší než procesor (stovky MHz), velikost stovky MB až GB, typicky dynamická paměť

Vnější paměť

- Sekundární paměti
 - Pevný disk, je na nich systém souborů (struktura adresářů), obsahuje obvykle statickou nebo dynamickou cache pro urychlení čtení/zápisu
- Terciární paměti
 - zařízení k zálohování dat, CD a DVD, Optické disky



Reprezentace záporných celých čísel

- **doplňkový kód – D(x)**
 - první bit je znaménkový
 - pro počítání se zápornými čísly není potřeba převádět čísla z doplňkového kódu

převod záporného čísla na kladné:

10111011 bitový obraz(8 bitů), 187 = -69 + 256

01000100 bitový doplněk

+1 přičtení jedničky

01000101 bitový obraz(8 bitů) 64 + 4 + 1 = 69

Reprezentace záporných celých čísel

- převod kladného na záporné:

01000101 bitový obraz(8 bitů), 69

10111010 bitový doplněk

+1 přičtení jedničky

10111010 bitový obraz(8 bitů)

přetečení: $127 + 1$: $01111111 + 00000001 = 10000000 \rightarrow -128$