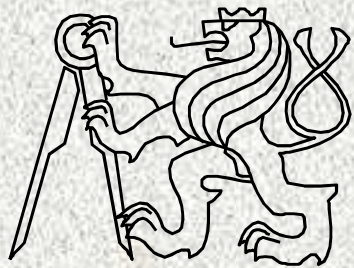
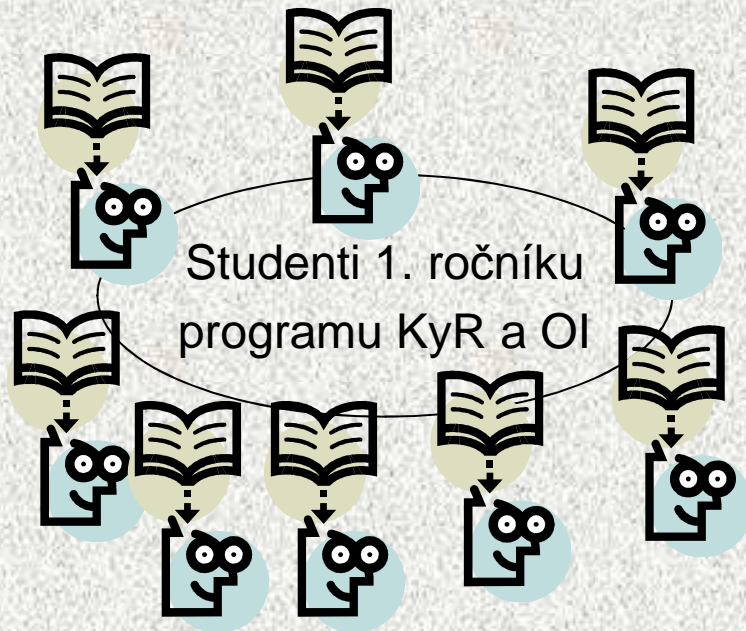


Programování 1 - Java



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. M. Filipský
- Ing. M. Balík
- Ing. M. Mudroch
- Ing. J. Tožička

- Garant: Doc.Ing. Jelínek Ivan CSc.



Proč
asi
😊??

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 datové a řídicí 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. Vstup a výstup, **řídící konstrukce**, větvení, cykly
3. **Struktura programu**, vývoj programu, ladění programu, vývojová prostředí, proměnné, výrazy, typy, logické a číselné operátory, první program.
4. **Koncepce Javy**, základní vlastnosti, současná podoba a vývoj, úvod do jazyka, zpracování programu, vnitřní forma, vývojová prostředí,
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. Rozklad problému na podproblémy, princip **rekurze a iterace**
8. **Třídy 1** - Principy objektového přístupu, třídy, třída jako programová jednotka, třída jako zdroj funkcí, třída jako datový typ, statické a instanční metody,
9. **Třídy 2** - zapouzdření, setry, getry, metody třídy Object -equals, toString, hashCode
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	max. bodů	min. bodů
10 domácích úloh	30	20
semestrální práce	24	12
test na cvičeních	16	8
zkouškový test	20	10
ústní zkouška	20	-10b

Minimální počet bodů pro zápočet je 40 bodů

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

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

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

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

Začínáme



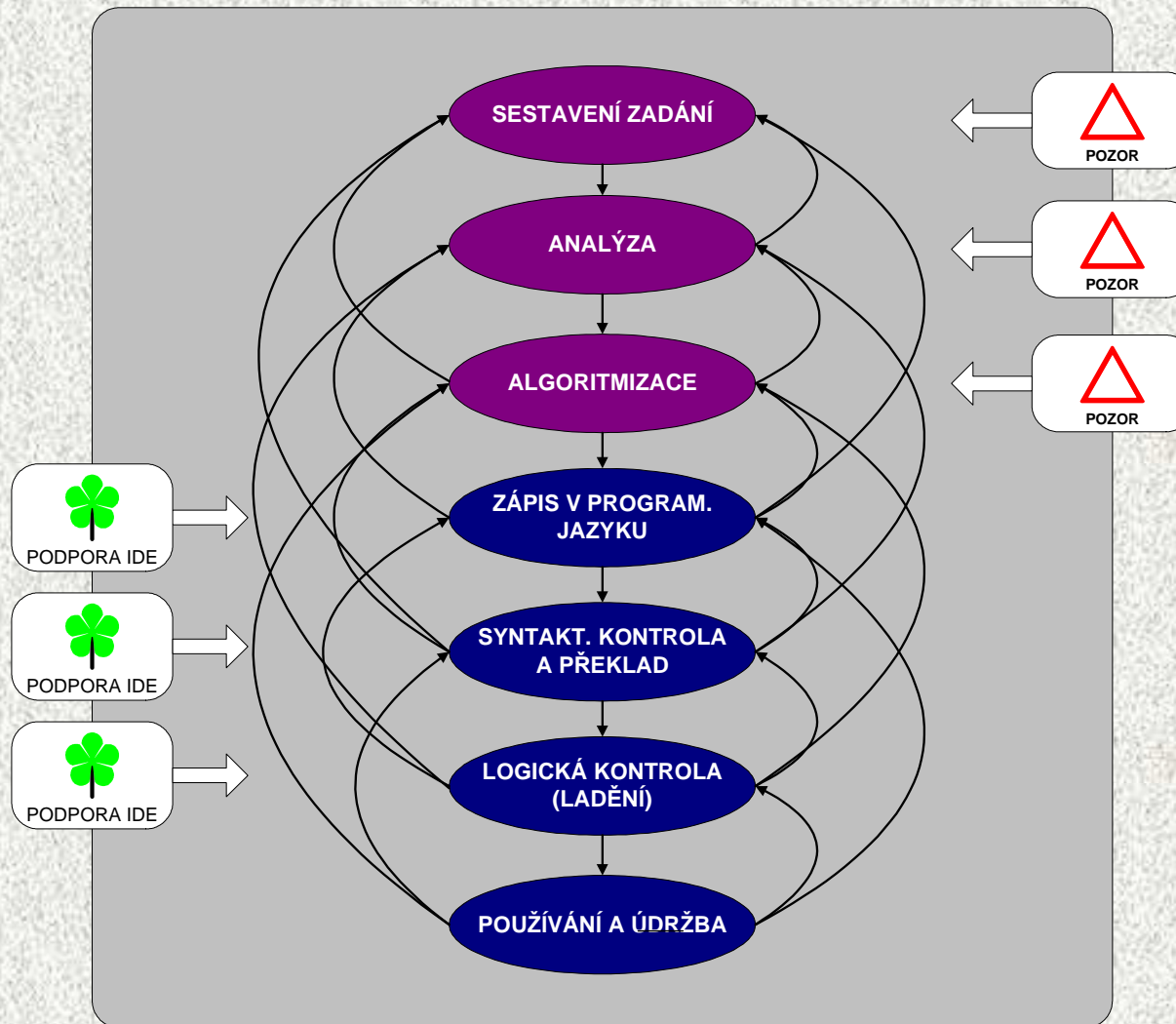
Šest zákonů programování

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



Proces programování



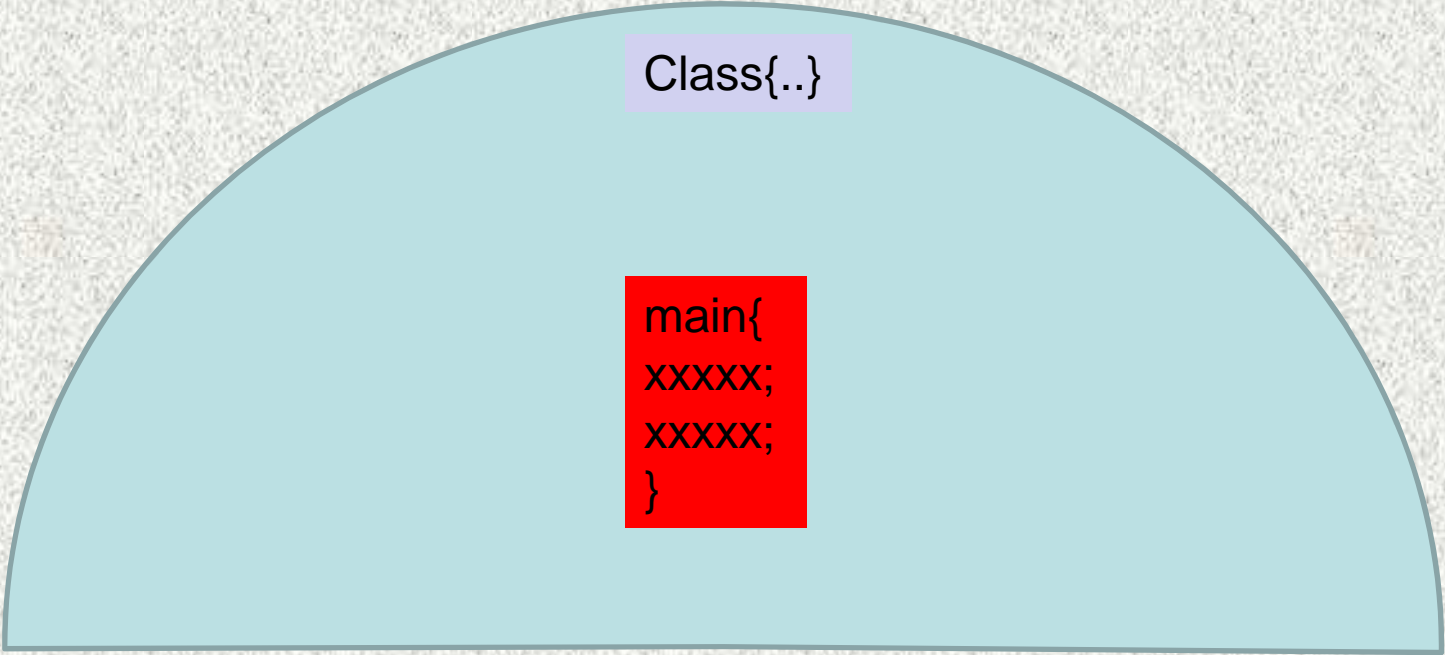
Programy a programovací jazyky

- 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

Základní přístupy k programování

- Strojový, naivní přístup
 - Strojový kód, assembler, skripty, C
- Procedurální přístup
 - C, Java, ...
- Objektový přístup
 - Java, C++, C#, ...

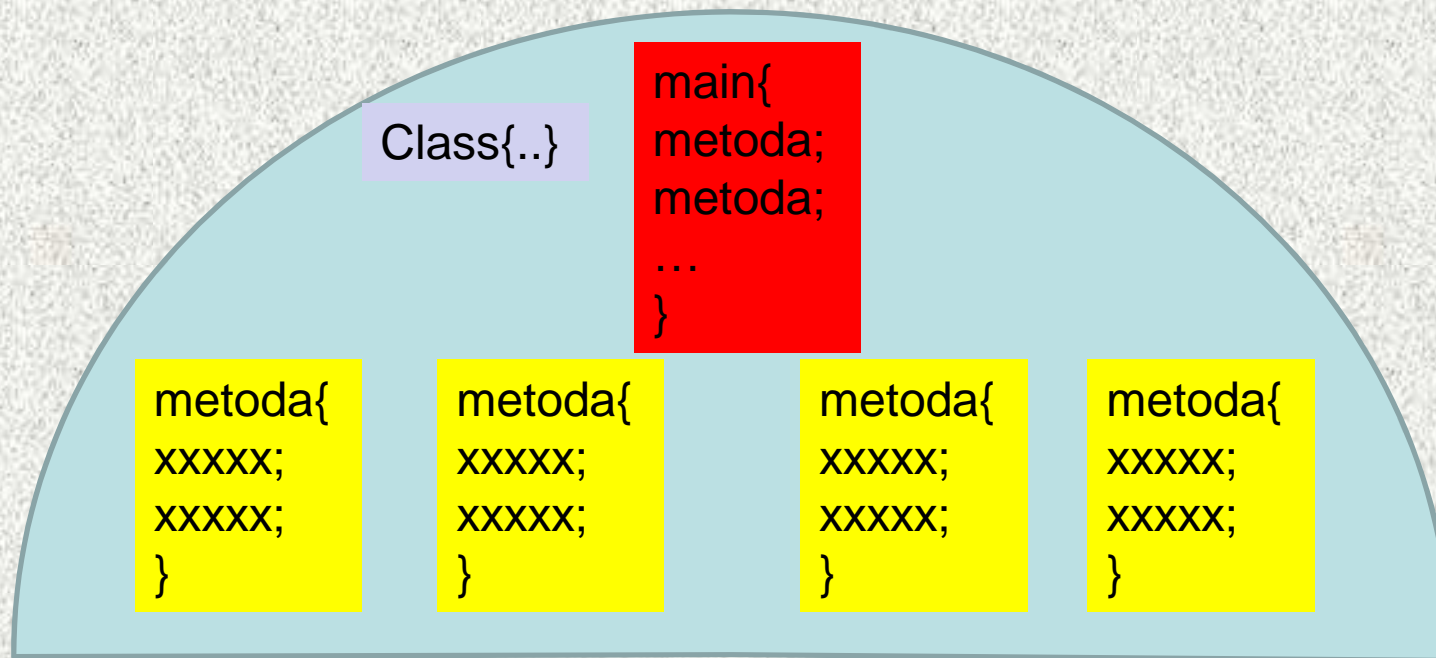
Naivní přístup v Javě



Class{..}

```
main{  
xxxxx;  
xxxxx;  
}
```


Procedurální přístup v Javě



- statické metody

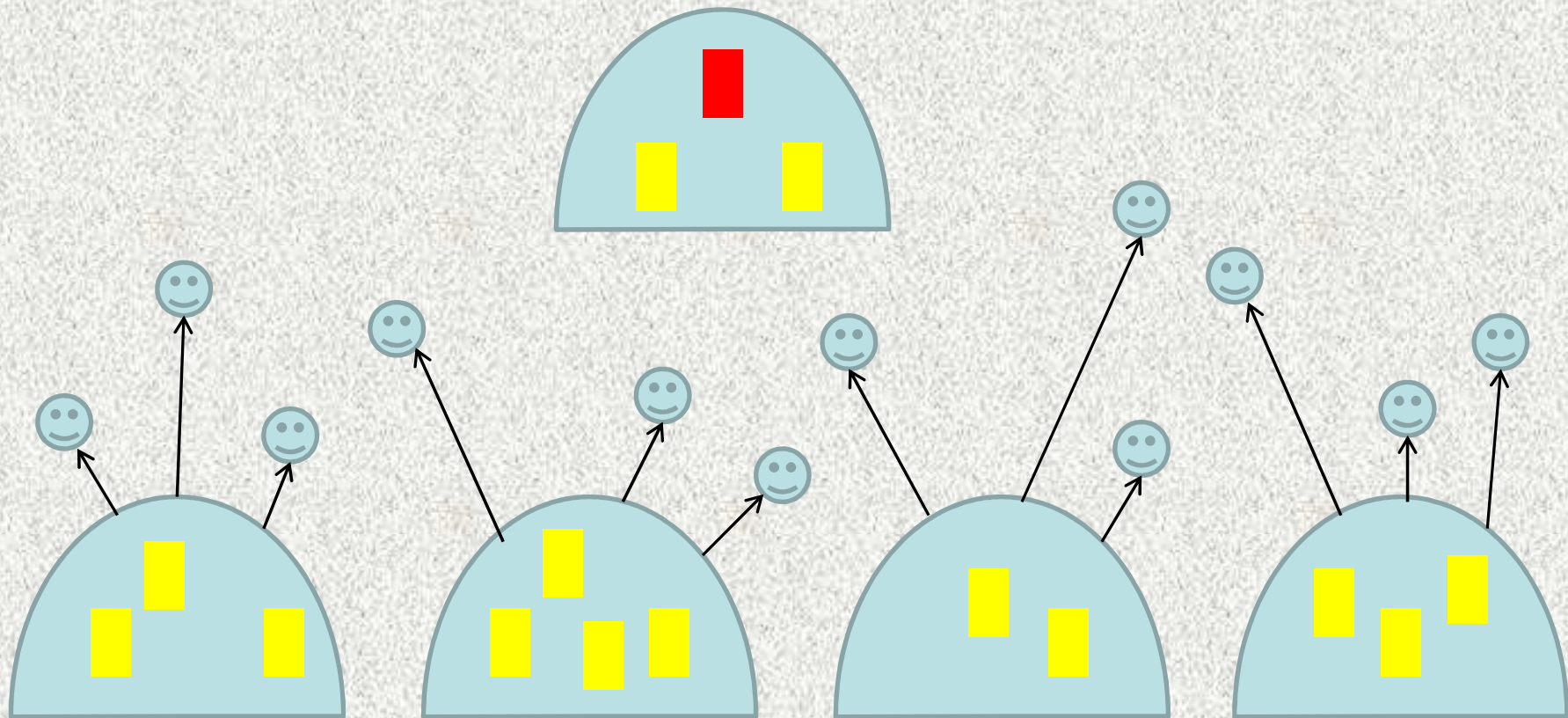
JAVA – procedurální přístup

```
public class Dekrementace {
```

```
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int n;  
        System.out.println("Zadej počet ");  
        n = sc.nextInt();  
        while (n > 0) {  
            System.out.println("Vypis " + n);  
            n = zmensi(n);  
        }  
    }
```

```
    static int zmensi(int n) {  
        n=n-1;  
        return n;  
    }
```

Objektový přístup v Javě



- instanční metody

JAVA – objektový přístup

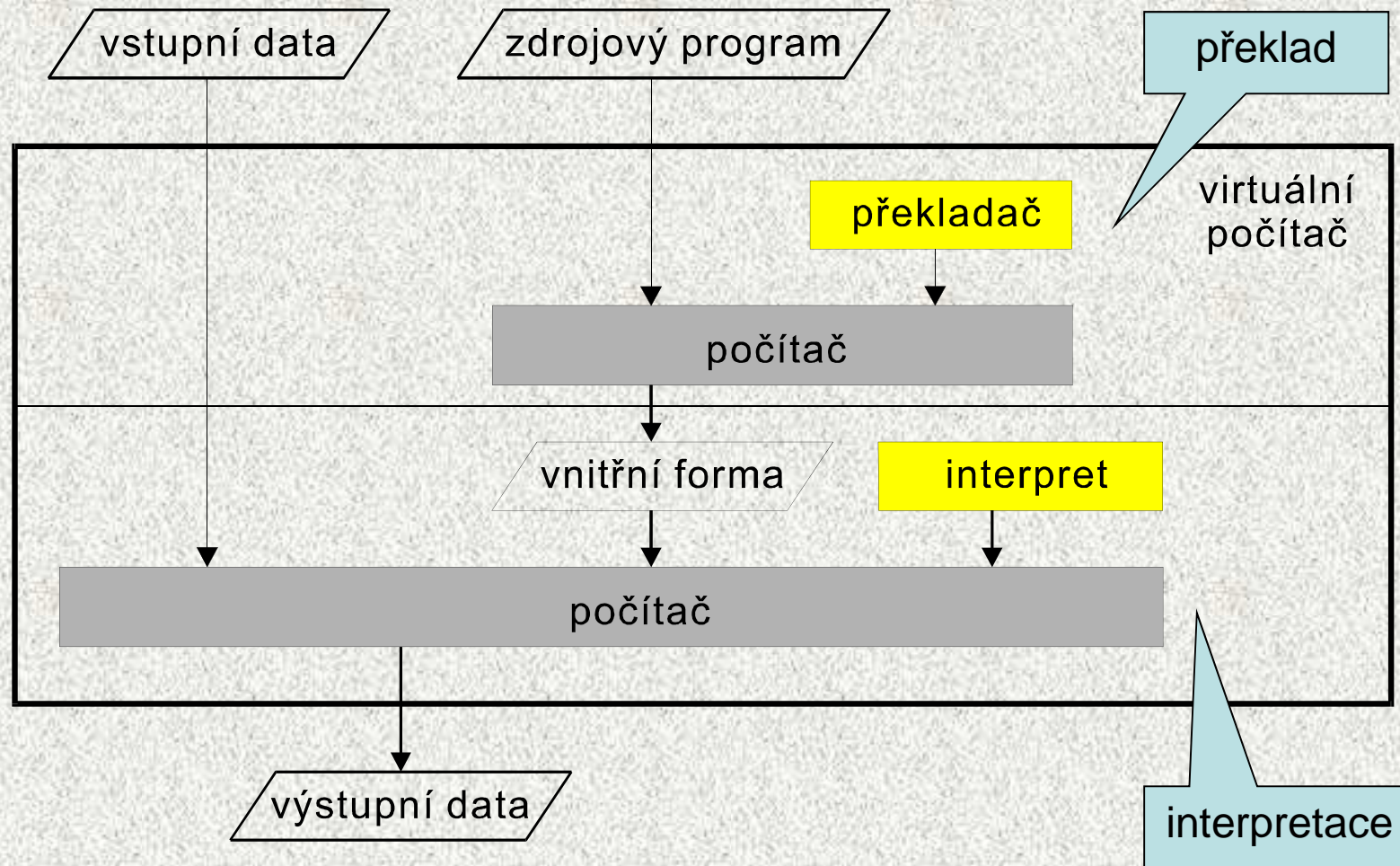
```
public class CitacTest{
public static void main(String[] args) {
Citac citac = new Citac(0);
Menu menu = new Menu();
do {
citac.zobraz("Hodnota = "+citac.hodnota());
switch (menu.vyber()) {
case 1: citac.zvetsit(); break;
case 2: citac.zmensit(); break;
case 3: citac.nastavit(); break;
}
} while (menu.volba()!=0);
citac.exit("Konec");
}
}
```

```
class Citac{
}
```

```
class Menu{
}
```


Implementace programovacích jazyků

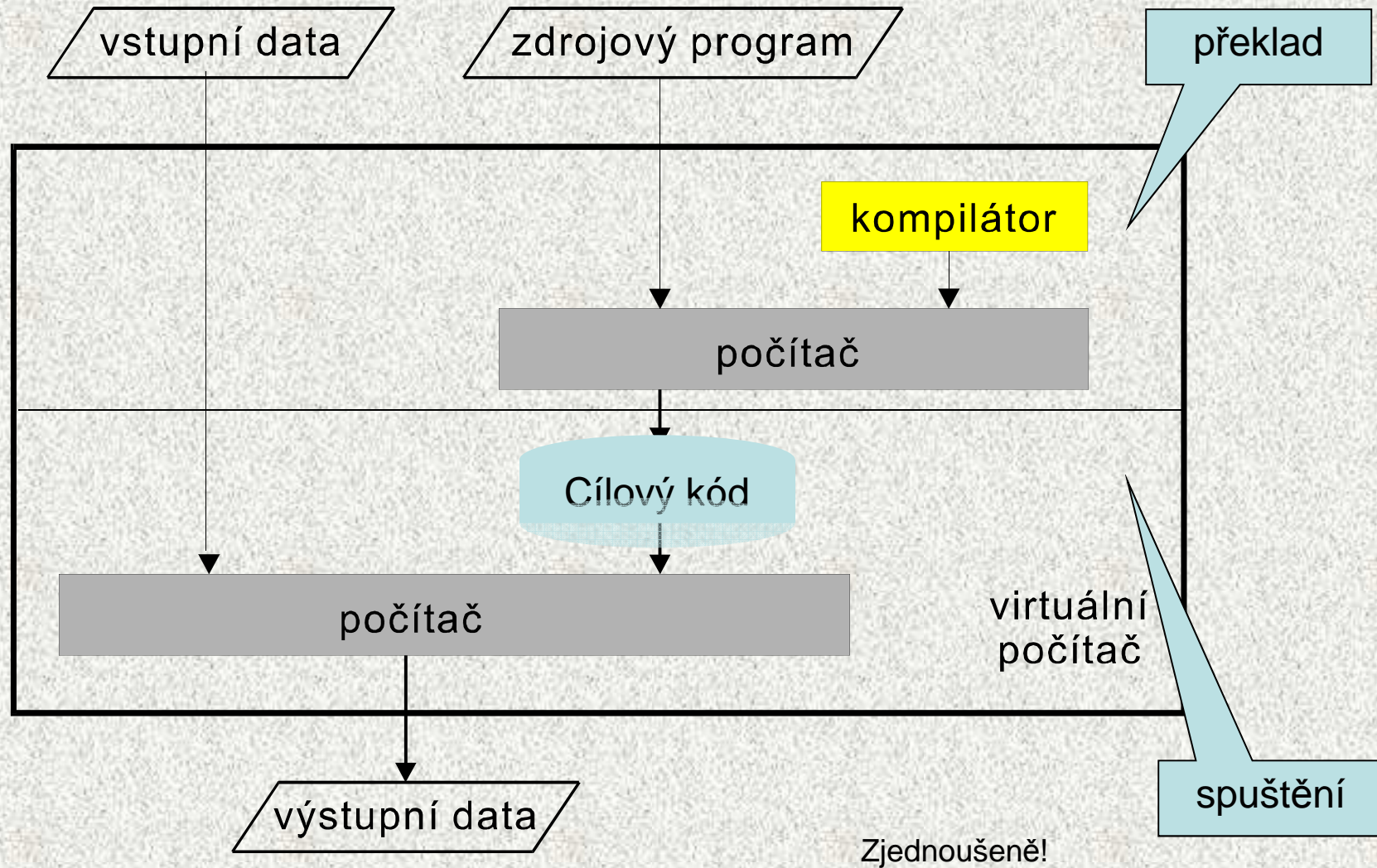
- Interpretační metoda:



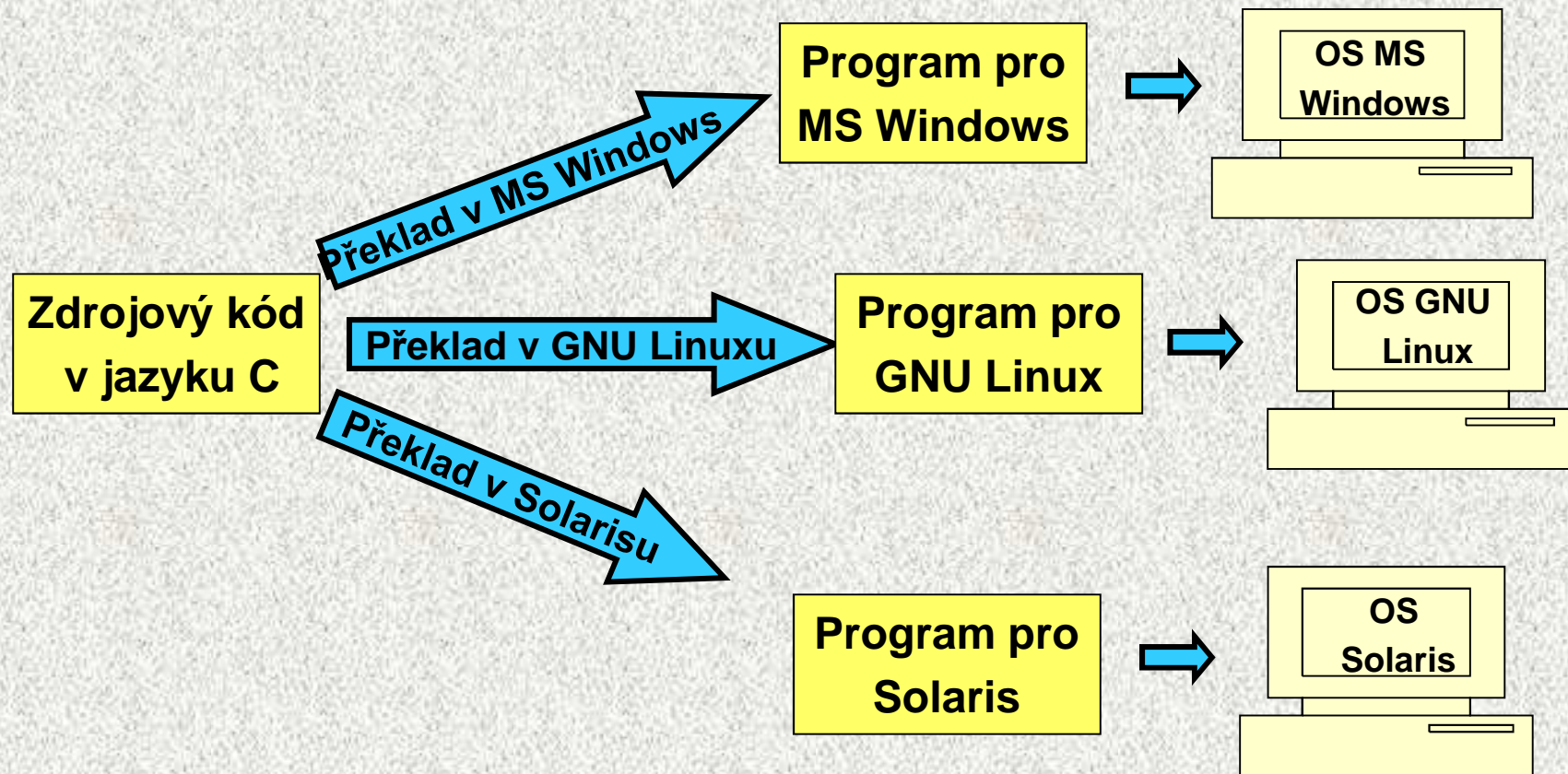
Zjednoušeně!

Implementace programovacích jazyků

- Kompilační metoda:



Kompilační metoda - jazyk C, C++

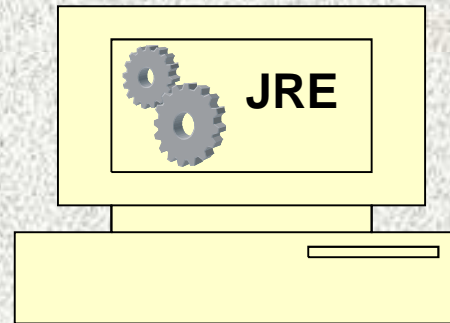
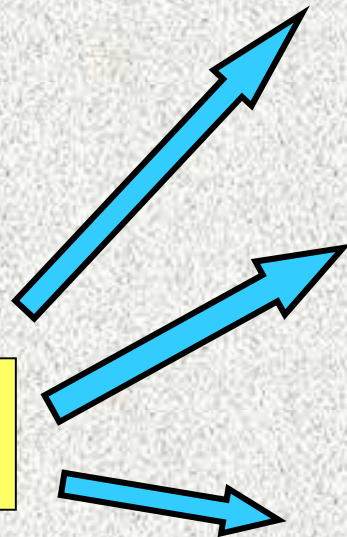


Interpretační metoda - jazyk Java

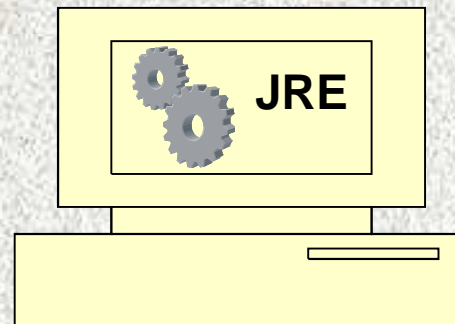
Zdrojový kód
v jazyku Java
soubor *.java*



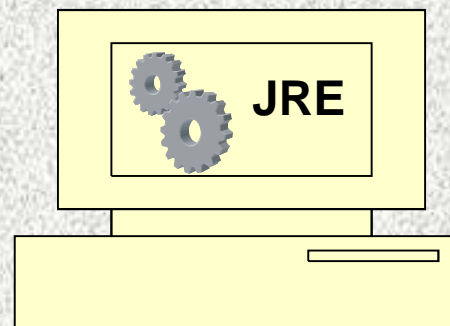
Bytecode
soubor *.class*



OS MS
Windows



OS GNU
Linux

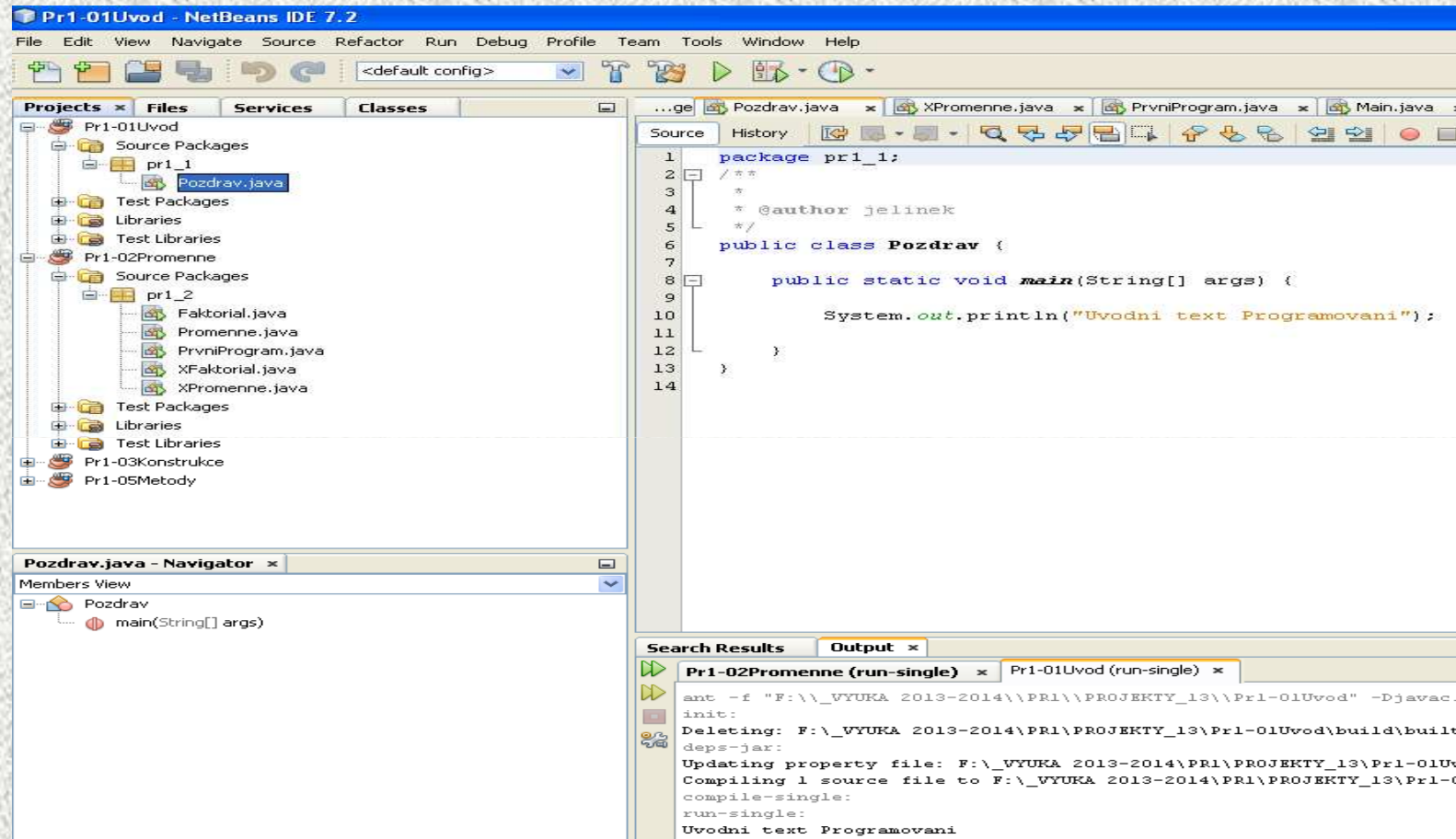


OS
Solaris

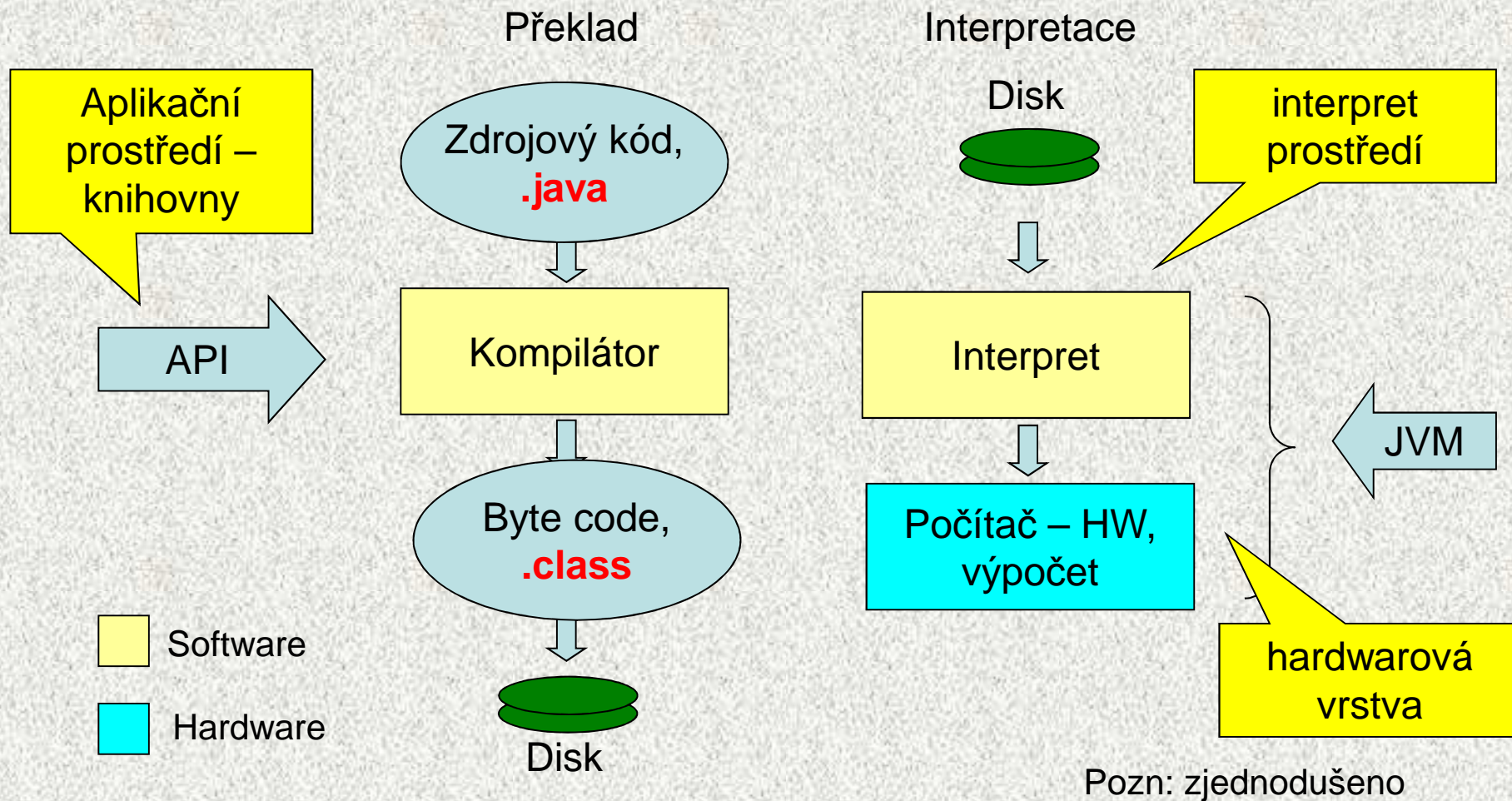
IDE – vývojový nástroj - NetBeans

- 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 + 2 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
- Jiná prostředí
 - Eclipse - opensource - zdarma, fa IBM
 - IDEA - komerční (30denní zkušební verze zdarma)
 - JBuilder - základní verze zdarma pro nekomerční využití, fa Borland
 - JDeveloper - vývojové prostředí firmy Oracle, freeware
 - BlueJ – bylo volně šiřitelné multiplatformní vývojové prostředí

NetBeans



Java Platforma (JRE) = Java Core API + JVM



Vývoj programů v Javě

- JRE - běhové prostředí, JRE = JVM +API
pro běh programů
 - JVM – Java Virtual Machine – virtuální stroj
 - API – Application Programming Interface - knihovny
- JDK - Software Development Kit pro Javu
 - sada základních nástrojů pro vývoj programů v Javě
 - JRE, překladač javac, debugger, javadoc, nástroje pro vytváření jar archivů, mnohé další
 - ke stažení na <http://java.sun.com>

Jazyk JAVA - interpretace

- 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!!!

Proč jazyk Java?

- jde o vyšší, obecně použitelný programovací jazyk s **vysokým stupněm zabezpečení**
- je **objektově orientovaný**, umožňuje však i klasické **procedurální programování**
- vytvořené programy jsou **zcela portabilní** (program vytvořený pod MS Windows bez problémů funguje pod Unixem a naopak)
- syntaxe výrazů a příkazů **vychází z jazyka C**; přechod z Javy na C nebo C++ je tedy jednodušší, než odjinud
- základní implementaci (JDK – Java Development Kit) firmy Sun lze pro prostředí Windows i Unix stáhnout ze stránek firmy Sun:
<http://java.sun.com>
- My používáme vývojové prostředí NetBeans 7., fy.
<http://www.netbeans.org/>
 - V současnosti je aktuální Netbeans 7.3.1 s Javou 7.0 u25
 - Studenti si tedy mohou snadno vývojový nástroj instalovat na svých domácích počítačích a mohou se učit programovat i mimo počítačové učebny školy

JAVA – první program

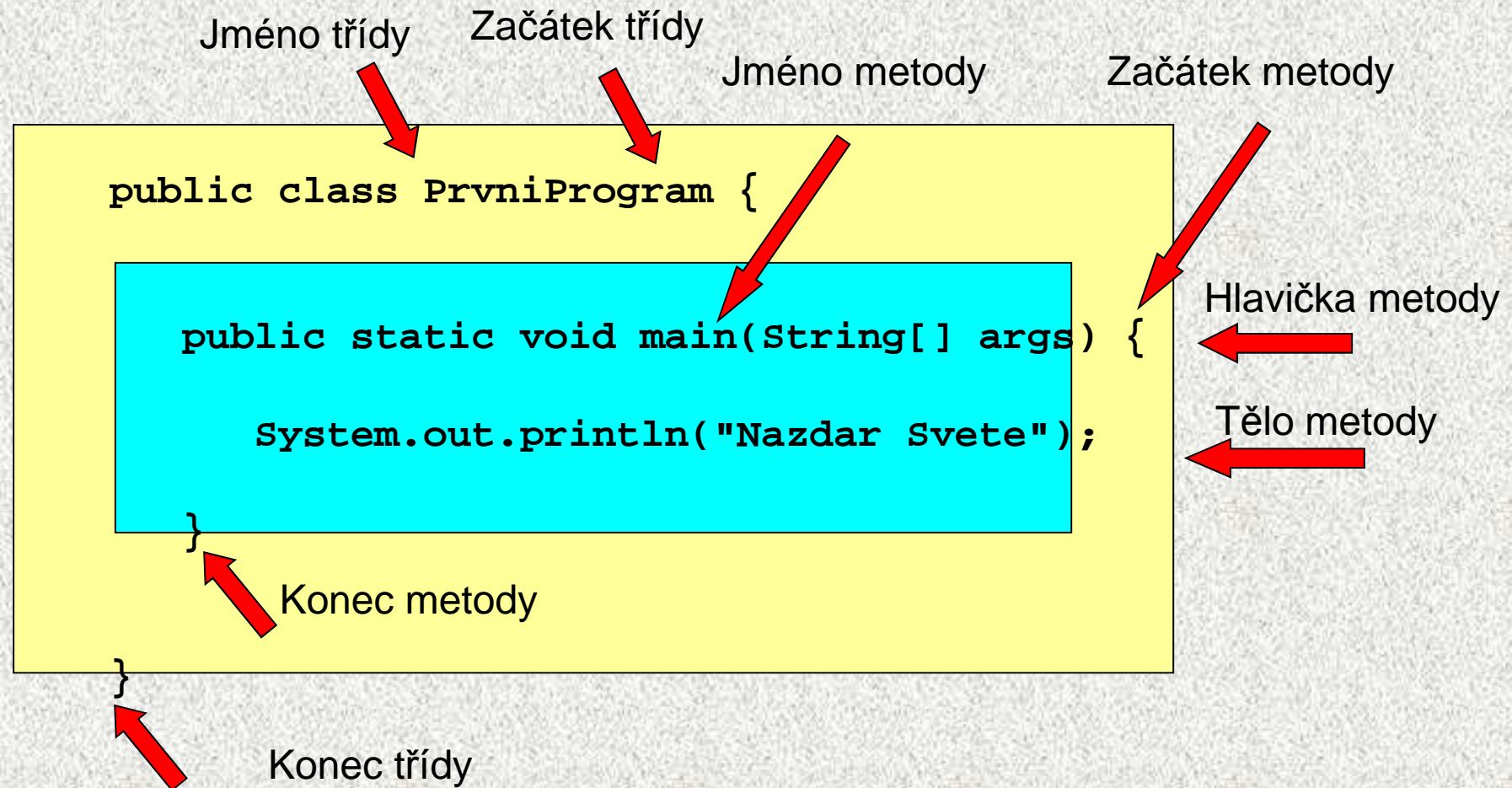
- Příklad: program vypíše daný text na obrazovku:

```
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

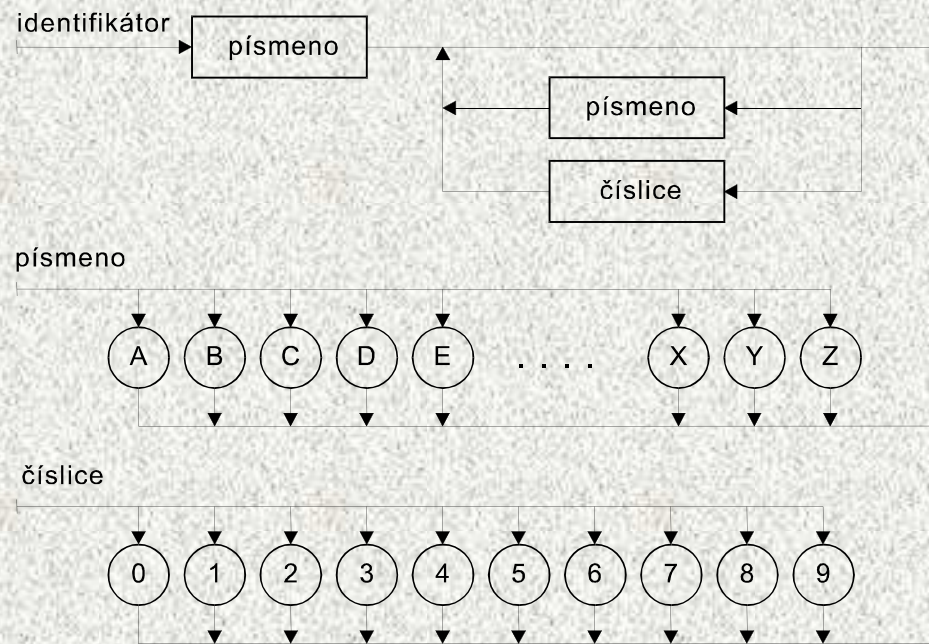
- Program má blokový charakter (blok třídy, blok(y) metod(y))
 - Nejtriviálnější program je tvořen metodou `main` ve třídě (třída = program)



Vlastnosti programovacích jazyků

- **Syntaxe**

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



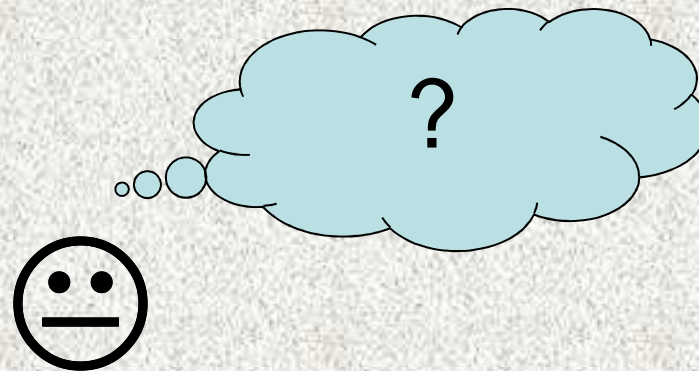
- **Sémantika**

- udává význam jednotlivých konstrukcí

Rozšířená BNF

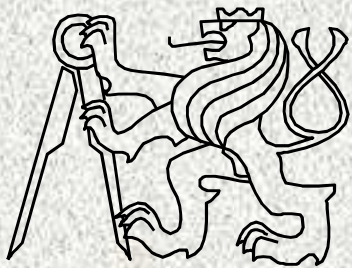
- Rozšířená Backus-Naurova forma – EBNF
- Příklad: identifikátor
 - identifikátor = písmeno {písmeno | číslice}
 - písmeno = 'A' | 'B' | 'C' | 'D' | ... | 'X' | 'Y' | 'Z'
 - číslice = '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'
- Neterminály:
 - identifikátor, písmeno, číslice
- Terminály: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

Vlastní studium



Programování 1 – Java

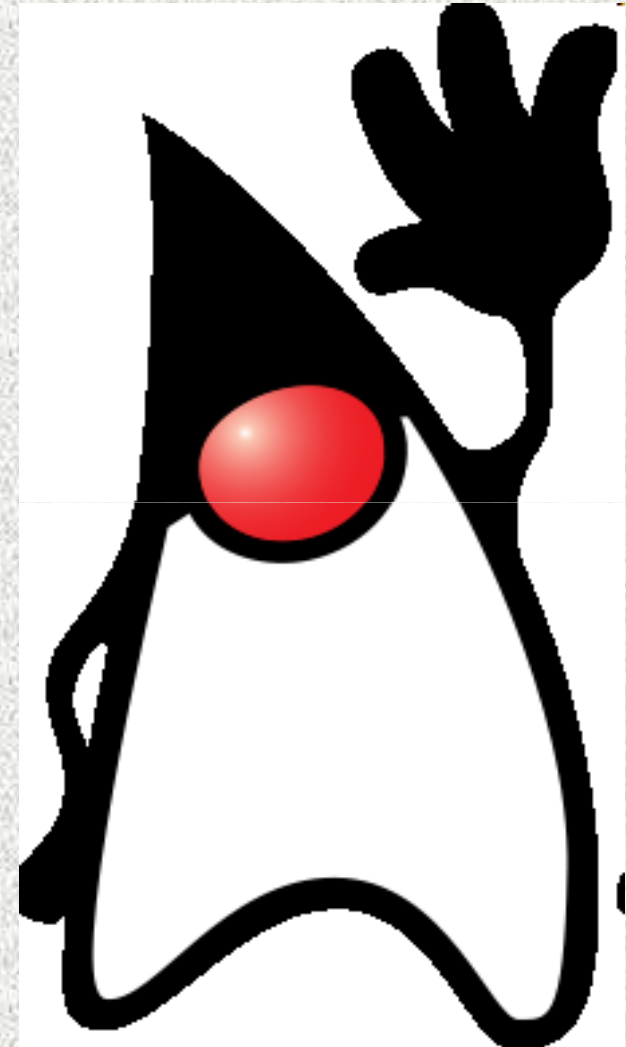
KONEC



A0B36PR1-Programování 1
Fakulta elektrotechnická
České vysoké učení technické

Java - historie

- v roce 1990 vznikl Green Team vedený Jamesem Goslingem – jazyk Oak (dub)
 - původně navržen pro vestavěné systémy
- Java představena firmou Sun Microsystems 23. května 1995
 - navržen pro vývoj aplikací na webu
- 1996 vydán první Java Development Kit 1.0 (pro aplety)
 - důležité využití
- 2008 - JDK 6 Update 13 s JavaFX SDK
- 2009 – JDK , 1.6.0_14;
- **2011 – JDK 7 !!!**
- **2012 – JDK 7.1 !!!**



Duke, maskot Javy

Java - edice

- každá edice je určena pro specifické účely, pro specifická zařízení
 - Java Card - smart (chytrý) karty - platební a kreditní karty
 - J2ME Java 2 Micro Edition – midlety, Kilobyte VM, mobilní zařízení, PDA, set-top boxy, vyžaduje 160 kb ROM a 32 kB RAM (CLDC, MIDP)
 - J2SE standard edition – stolní počítače
 - J2EE enterprise edition – podnikové aplikace