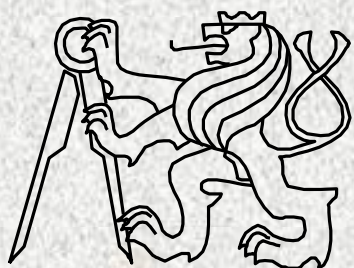
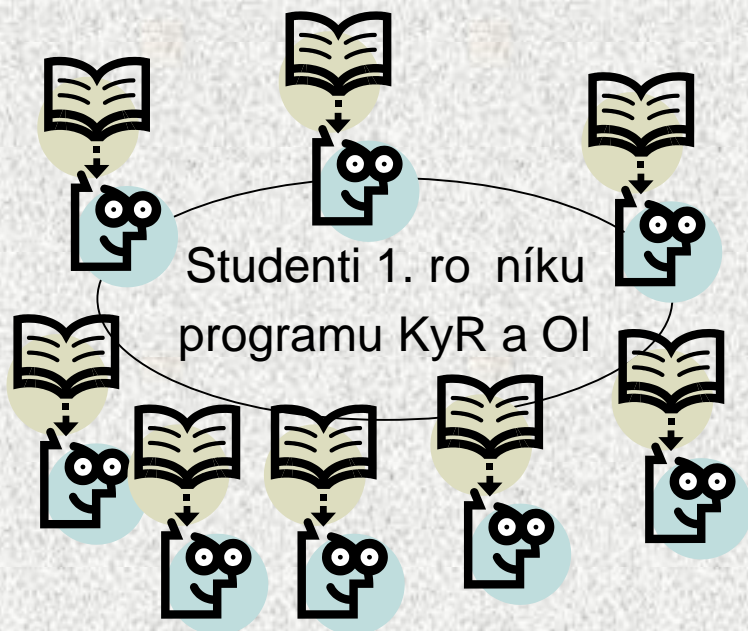


Programování 1 - úvod



A0B36PR1-Programování 1
Fakulta elektrotechnická
eské vysoké u ení technické

Učíte, představme se Ā ☺



Přednázející
Doc. Ing. Ivan Jelínek, CSc.

Cvičící: ě .



- “ Jste STUDENTI VUT FEL!
- “ Jste členy akademické obce!
- “ Dekan, senát, vedoucí rada
- “ Zaměstnanci (tituly ě), doktorandi
- “ Katedry

Je kdo v Programování 1

1. Vy . studenti

“ R zné zkuzenosti s r znými programovacími jazyky

2. P ednázející:

“ Doc.Ing. Jelínek Ivan CSc.

3. Nejd le0it jzí u itelé: cvi íci!

“ Ing. Buk Zden k

“ Ing. M. Filipský

“ Ing. M. Balík

“ Ing. M. Mudroch

“ Ing. J. To0i ka

“ Garant: Doc.Ing. Jelínek Ivan CSc.



Pro
asi
☺??

Způsob záznamu a hodnocení předmětu

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

(*)Způsob záznamu předmětu:
- zápočet
- klasifikovaný zápočet
- zápočet / zkouška

Cíl p edm tu

“ Programování 1

- “ reprezentace dat v počítači
- “ reprezentace čísel
- “ základní datové a řídicí struktury jazyka Java
- “ cykly
- “ jednoduché programy v Jav
- “ ladění program
- “ procedurální vs. objektový přístup
- “ objekty, třídy
- “ soubory a proudy, kolekce,
- “ ∞

- “ prekvizita Programování 2
- “ prekvizita Algoritmizace

Osnova p ednázek

1. **Základní pojmy** výpo etní techniky, opera ní systém, software, p eklada , interpret, programovací jazyky, syntaxe, sémantika
2. **Koncepce Javy**, základní vlastnosti, sou asná podoba a vývoj, úvod do jazyka, zpracování programu, vnit ní forma, vývojová prost edí,
3. **Struktura programu**, vývoj programu, lad ní programu, vývojová prost edí, prom nné, výrazy, typy, logické a íselné operátory, první program.
4. Vstup a výstup, **ídící konstrukce**, v tvení, cykly
5. **Funkce**, procedury, parametry, statické prom nné, lokální prom nné, blok, princip p id lování pam ti prom nným, halda, zásobník
6. **Pole**, referen ní prom nná typu pole, pole jako parametr, funkce typu pole.
7. **T ídy 1** - Principy objektového p ístupu, t ídy, t ída jako programová jednotka, t ída jako zdroj funkcí, t ída jako datový typ, statické a instan ní metody,
8. **T ídy 2** - zapouzd ení, setry, getry, metody t ídy Object -equals, toString, hashCode
9. Rozklad problému na podproblémy, princip **rekurze a iterace**
10. **Spojové 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 byt , úvod do zpracování výjimek, ukládání/ tení primitivních typ , primitivních typ a objekt (et zc), objekt do souboru Ě serializace;
13. Základní principy **vyhledávání, azení**
14. Rezerva

Osnova cvičení

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



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.: Účebnice 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 bod	max. bod	min. bod
4 domácí úlohy	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

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

Možnost nechat si zapsat známku nebo jít k ústní zkoušce. Pokud odteče 10 bodů

Zakonění podmínku: zápočet, **zkouška** (na základě bodového hodnocení)

Za ínáme



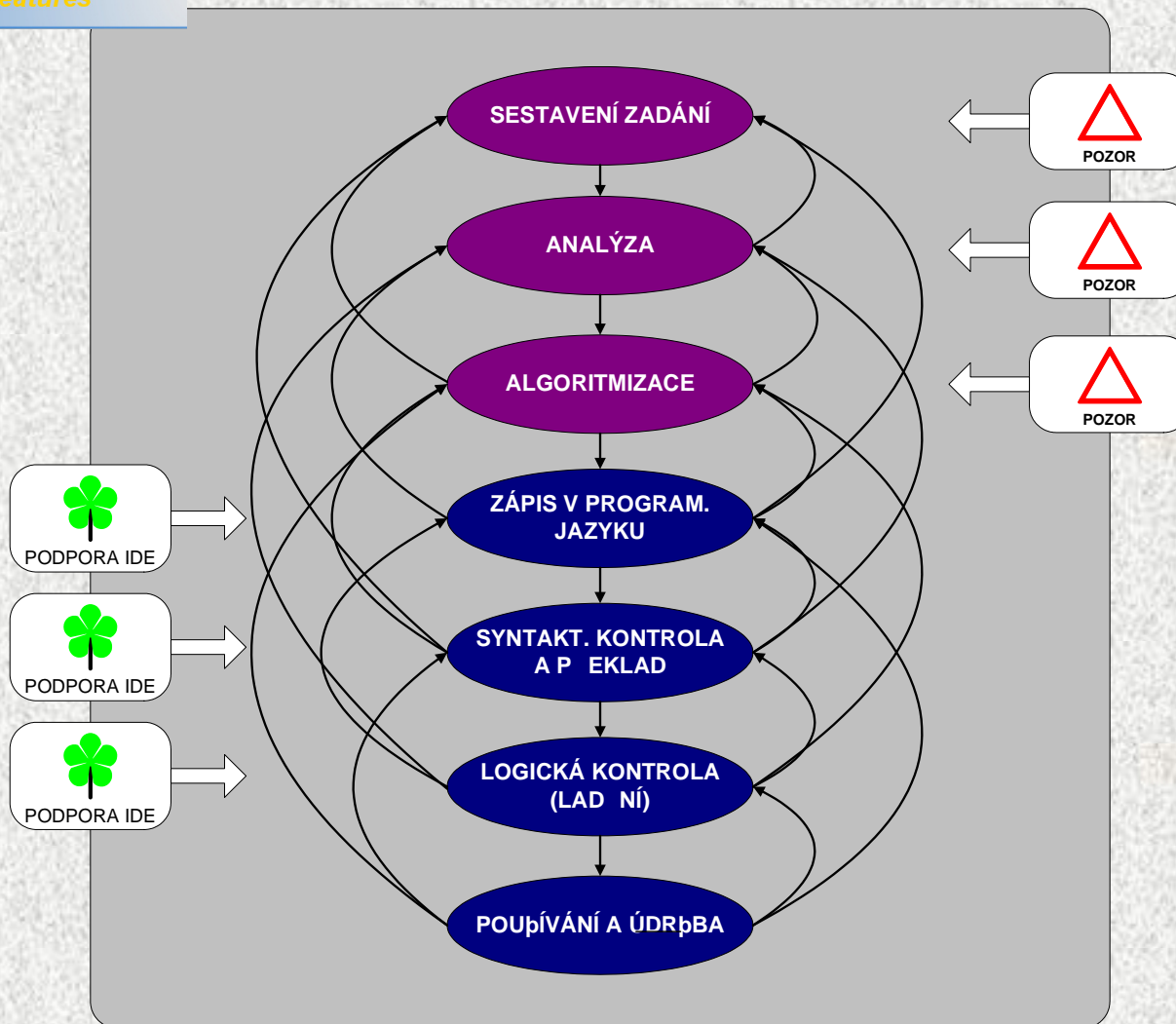
šest zákonů programování

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

Za ínáme doopravdy



Proces programování



Algoritmy 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 způsobí výpočty

dní p ístupy k programování

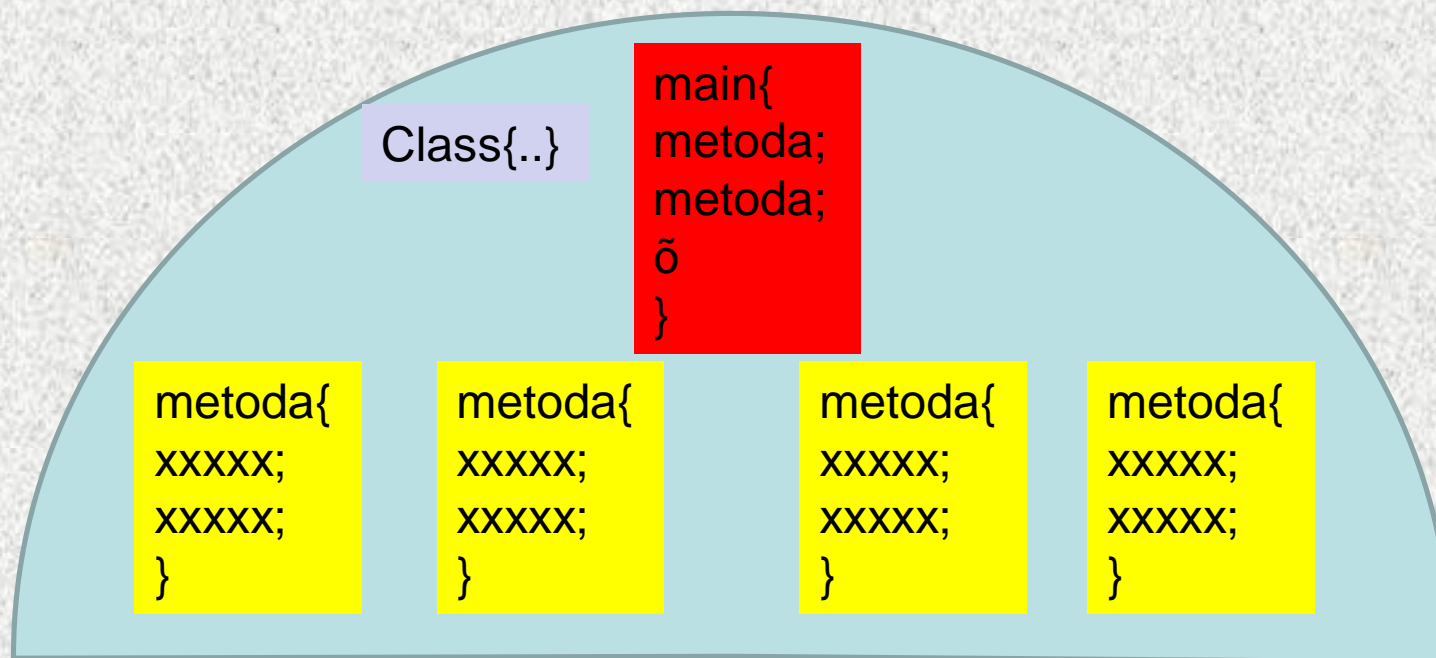
- “ Strojový, naivní p ístup
 - . Strojový kód, assembler, skripty, C
- “ Procedurální p ístup
 - . C, Java, ò
- “ Objektový p ístup
 - . Java, C++, C#, ò

Hlavní přístup v Jav

Class{..}

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


edurální p ístup v Jav



“ statické metody

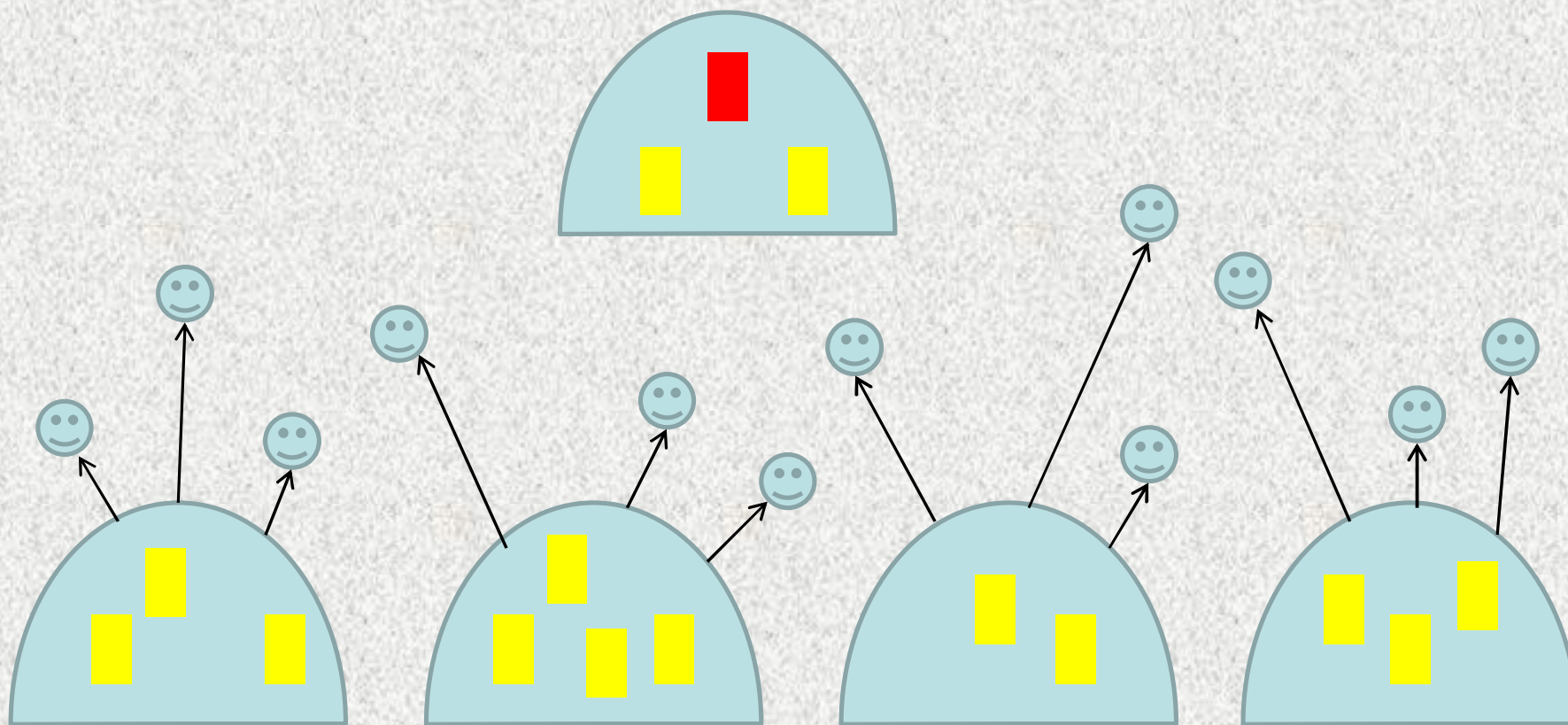
VA . procedurální p ístup

```
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 Jav



” instancí metody

AVA . objektový p ístup

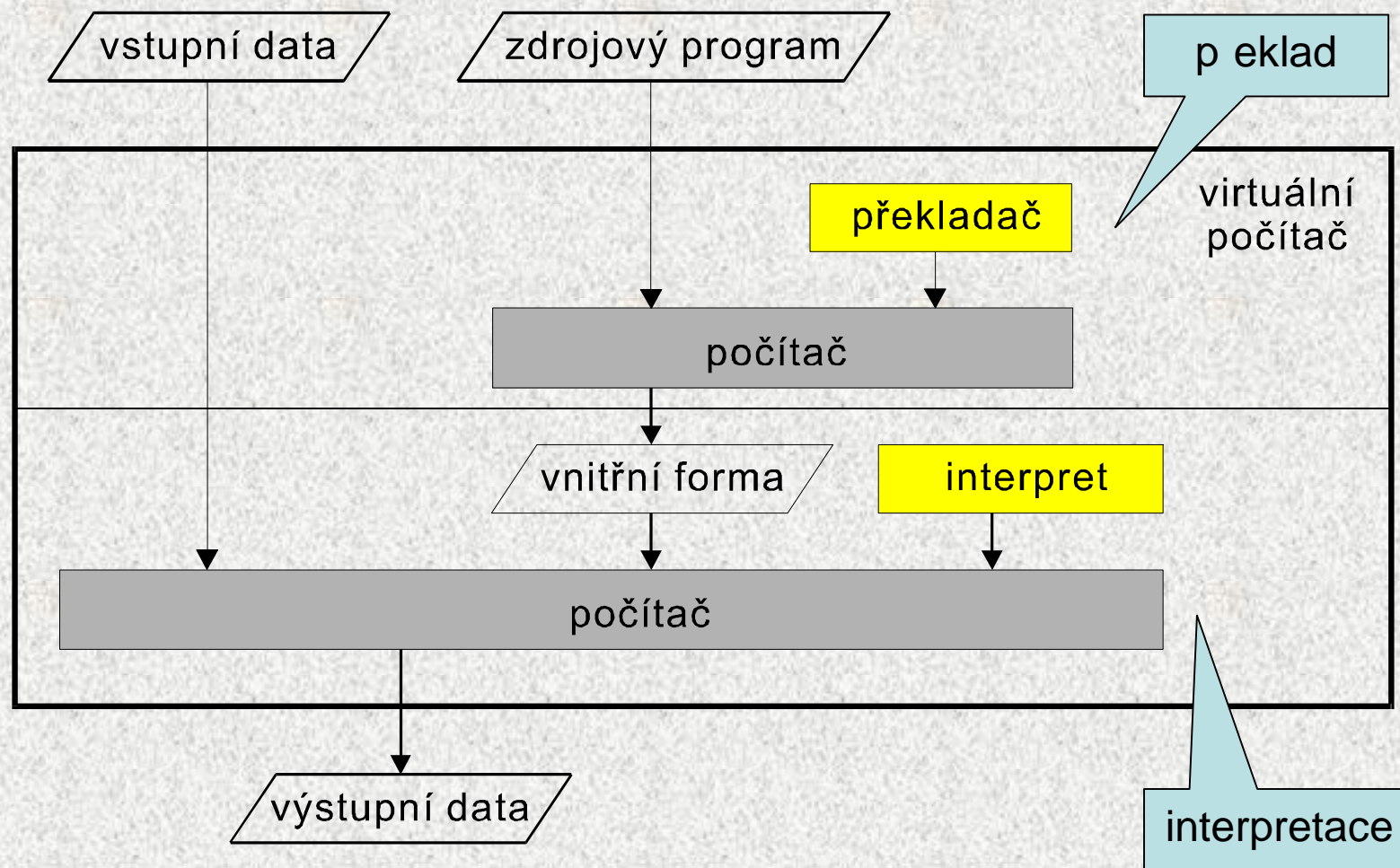
```
public class Citac_Test{  
public static void main(String[] args) {  
Citac citac = new Citac(0);  
Menu menu = new Menu();  
do {  
citac.zobraz("Hodnota = "+citac.hodnota());  
switch (menu.vyber()) {  
case 1: citac.zvetsit(); break;  
case 2: citac.zmensit(); break;  
case 3: citac.nastavit();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

```
class Citac{  
}
```

```
class Menu{  
}
```

Implementace programovacích jazyk

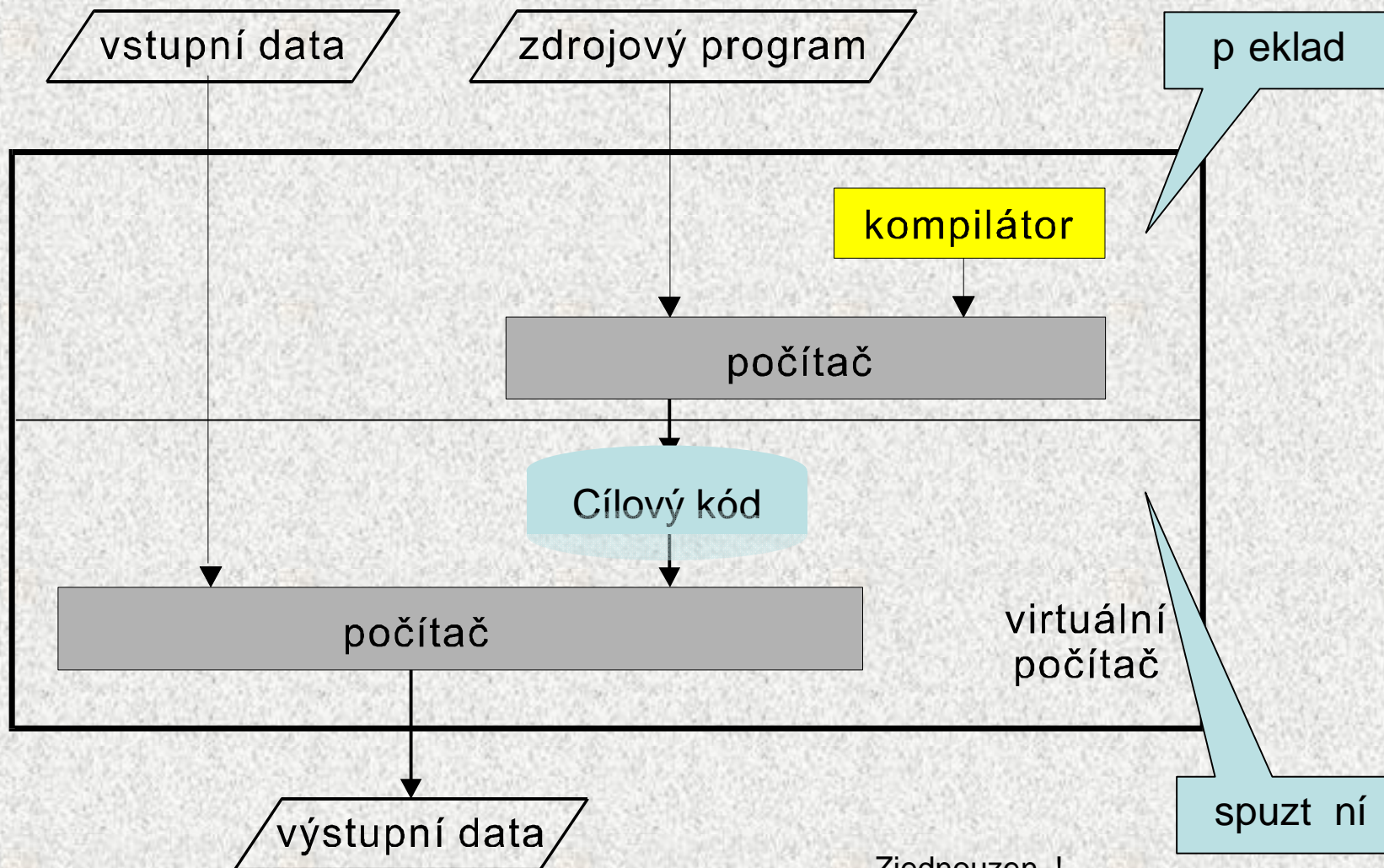
Interpretace metoda:



Zjednouzen !

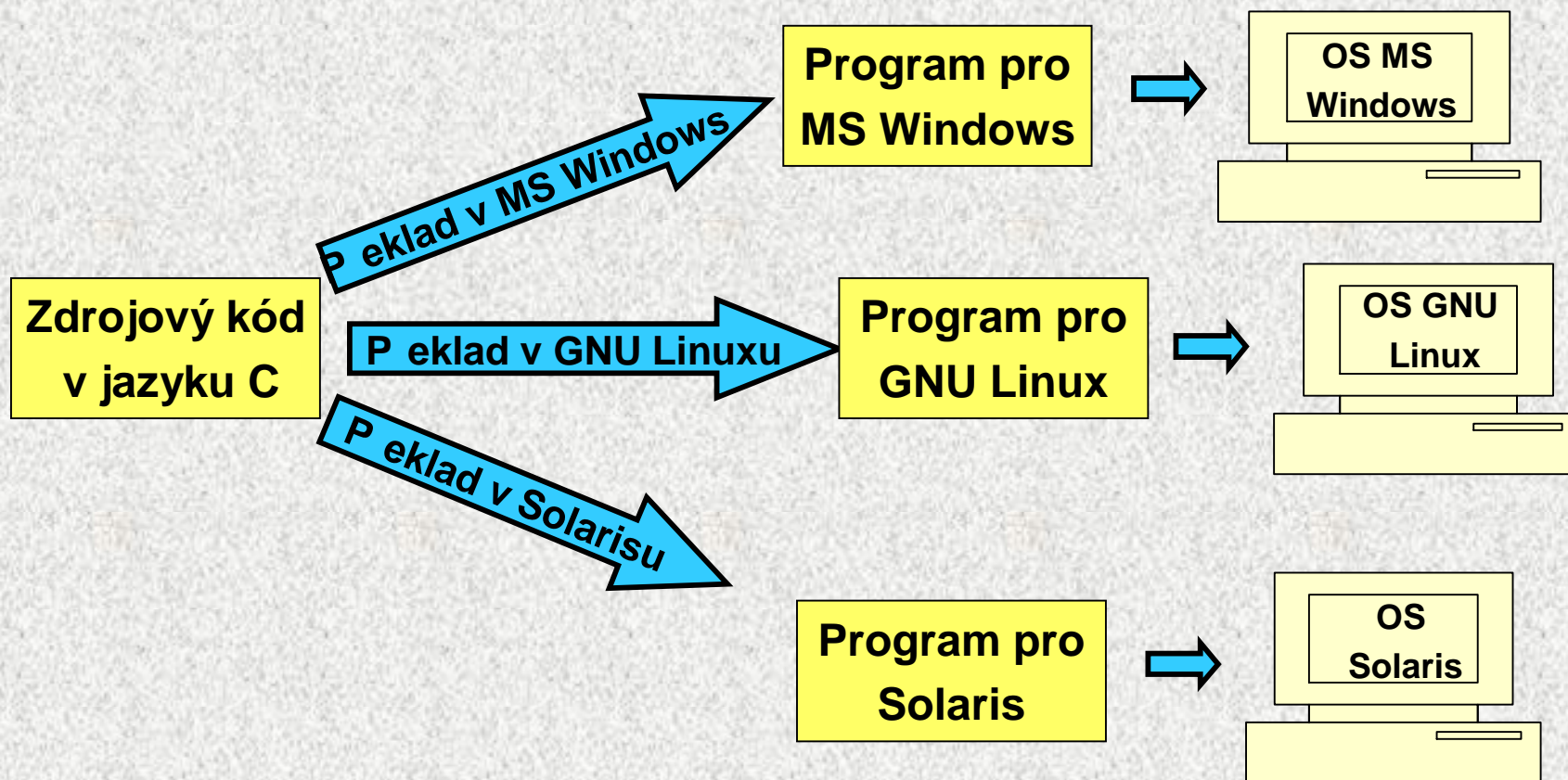
Implementace programovacích jazyk

” Kompilací metoda:



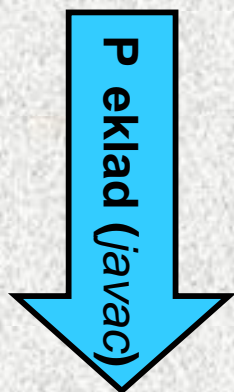
Zjednouzen !

Praktická aplikace - jazyk C, C++

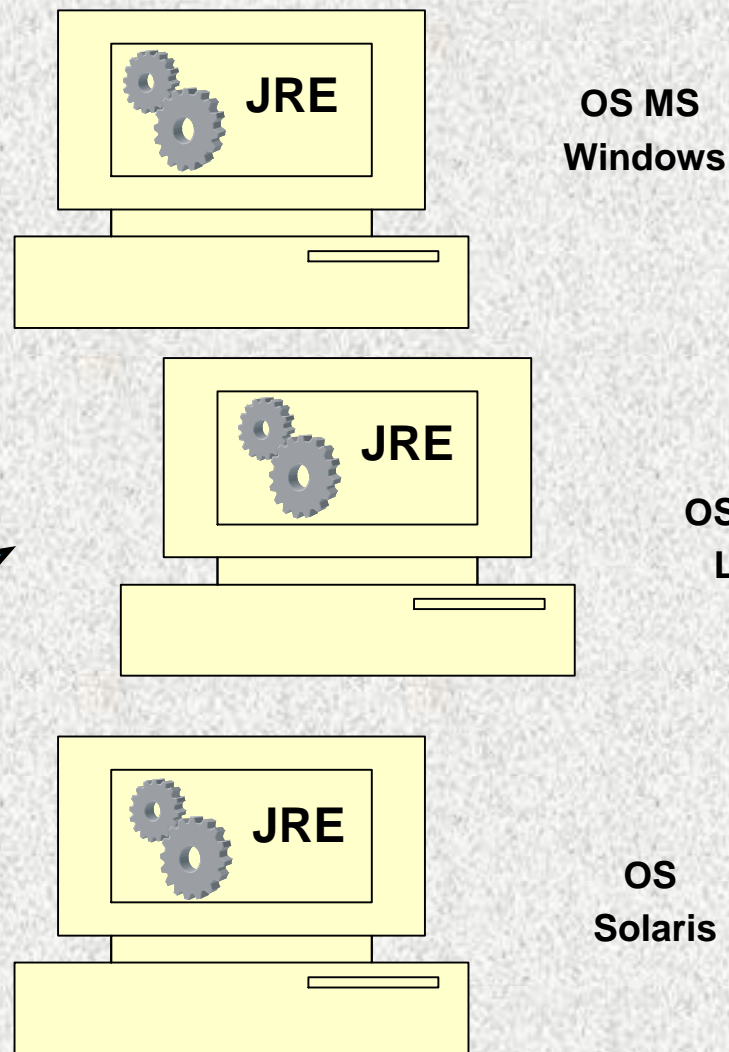


eta ní metoda - jazyk Java

Zdrojový kód
v jazyku Java
soubor `.java`



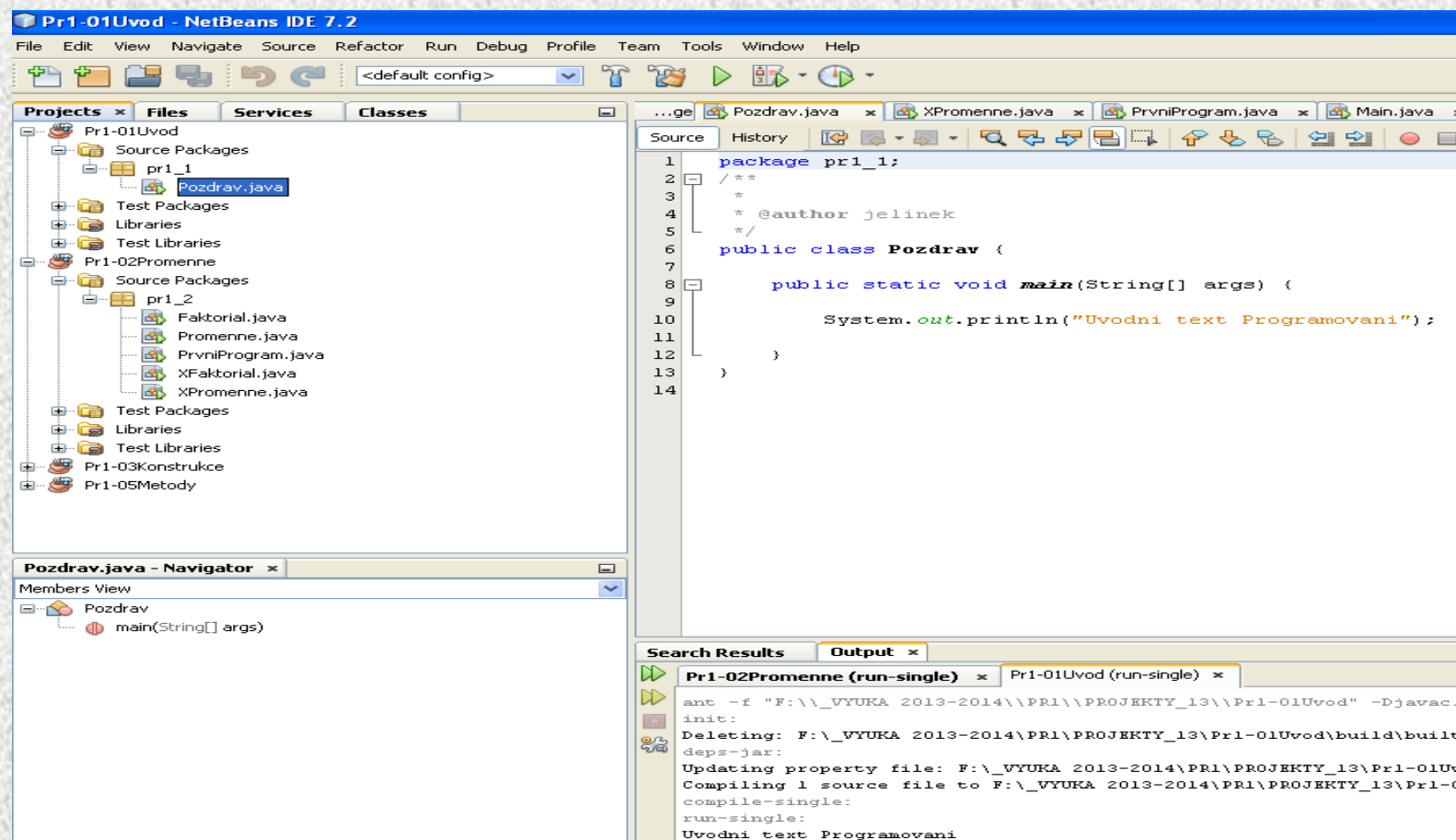
Bytecode
soubor `.class`



• vývojový nástroj - NetBeans

- “ Pro vývoj programů se používá vývojový nástroj nazývaný IDE (Integrated Development Environment)
- “ IDE je v dnešní době k dispozici pro všechny běžně používané programovací jazyky, často od různých výrobců software
- “ V předmetu 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 prostředí a další části)
- “ S používáním NetBeans se seznámíte na cvičeních
- “ Jiná prostředí
 - Eclipse - opensource - zdarma, od IBM
 - IDEA - komerční (30denní zkušební verze zdarma)
 - JBuilder - základní verze zdarma pro nekomerční využití, od Borland
 - JDeveloper - vývojové prostředí firmy Oracle, freeware
 - BlueJ - bylo volně dostupné multiplatformní vývojové prostředí

NetBeans



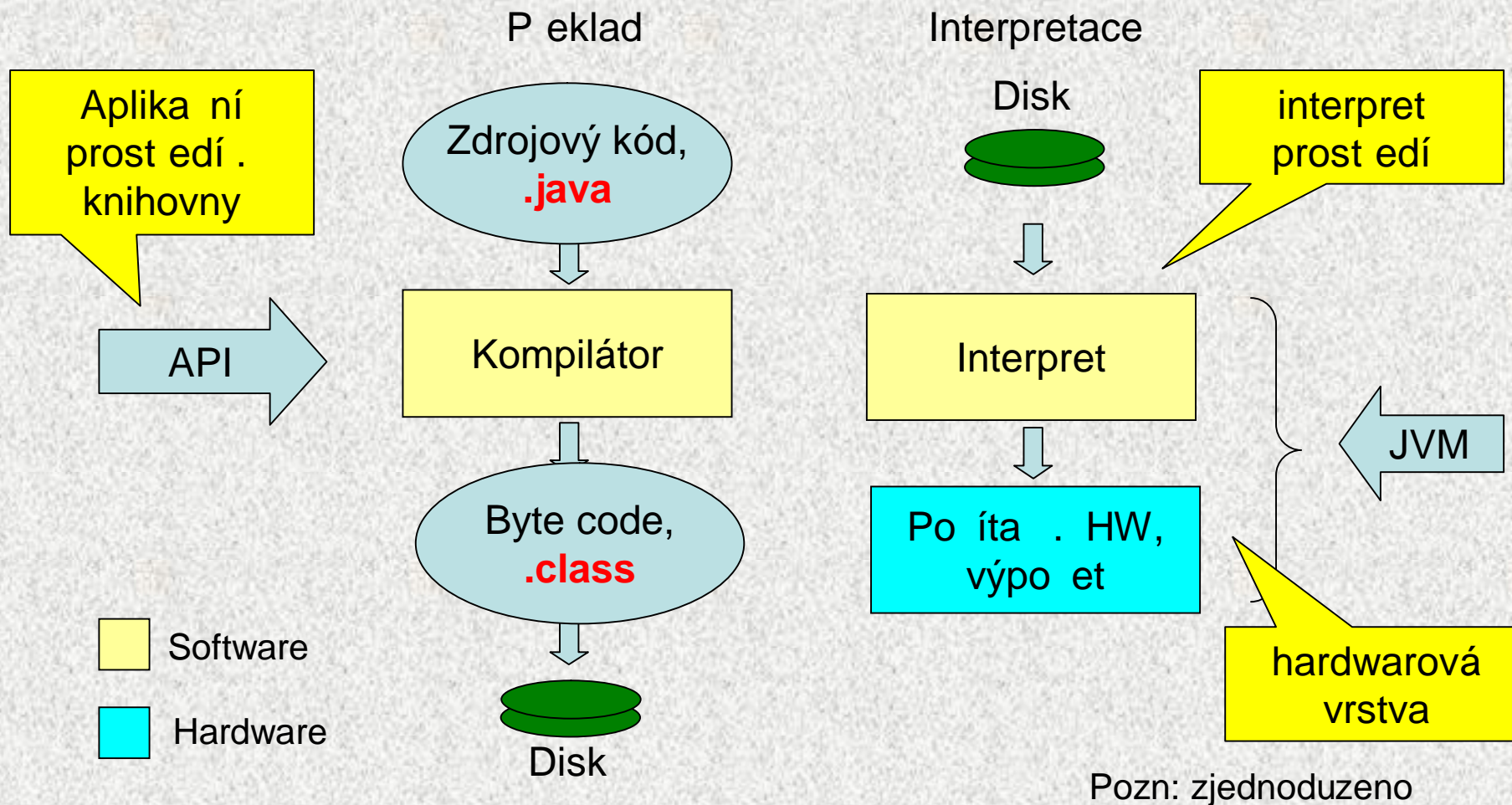
The screenshot displays the NetBeans IDE 7.2 interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The toolbar contains various icons for file operations and execution. The left sidebar shows a project tree for 'Pr1-01Uvod' with sub-packages 'pr1_1' and 'pr1_2'. The 'pr1_1' package contains 'Pozdrav.java'. The 'pr1_2' package contains 'Faktorial.java', 'Promenne.java', 'PrvniProgram.java', 'XFaktorial.java', and 'XPromenne.java'. The main editor window shows the source code of 'Pozdrav.java' with the following content:

```
1 package pr1_1;
2 /**
3  *
4  * @author jelinek
5  */
6 public class Pozdrav {
7
8     public static void main(String[] args) {
9
10         System.out.println("Uvodni text Programovani");
11     }
12 }
13
14
```

The bottom right pane shows the 'Output' window with the following text:

```
Pr1-02Promenne (run-single) x Pr1-01Uvod (run-single) x
ant -f "F:\_VYUKA 2013-2014\PR1\PROJEKTY_13\Pr1-01Uvod" -Djavac.
init:
Deleting: F:\_VYUKA 2013-2014\PR1\PROJEKTY_13\Pr1-01Uvod\build\build
deps-jar:
Updating property file: F:\_VYUKA 2013-2014\PR1\PROJEKTY_13\Pr1-01Uv
Compiling 1 source file to F:\_VYUKA 2013-2014\PR1\PROJEKTY_13\Pr1-0
compile-single:
run-single:
Uvodni text Programovani
```

Java (JRE) = Java Core API + JVM



Vývoj program v Jav

- “ JRE - obsahuje prostředí, JRE = JVM +API
pro běh program
 - . JVM . Java Virtual Machine . virtuální stroj
 - . API . Application Programming Interface - knihovny

- “ JDK - Software Development Kit pro Javu
 - sada základních nástroj pro vývoj program v Jav
 - JRE, příkazy javac, debugger, javadoc, nástroje pro vytváření jar archivů, mnohé další
 - ke stažení na <http://java.sun.com>

Jazyk JAVA - interpretace

- “ Jazyk Java je implementován **interpret** ním zp sobem
 - “ program je tvo en jedním nebo n kolika **zdrojovými soubory** s p íponou **.java**:
`Program.java`
 - “ zdrojové soubory se p eloží p eklada em(*) **javac** do **vnit ní formy** (byte code, bajt-kód) s p íponou **.class**:
`Program.java > javac > Program.class`
 - “ **interpretaci** vnit ní formy **provede program java** (JVM . Java Virtual Machine v balí ku **JRE** Java Runtime Environment) a provede výpo et:
`Program.class > java > „výpočet“`

Poznámky:

(*) v terminologii firmy Sun to je kompilátor

- “ program obvykle **využívá** **adu knihoven (Java Core API)**, které je t eba mít k dispozici jak p í ekladu, tak p í interpretaci!!!

Pro jazyk Java?

je velmi pou0itelný programovací jazyk s vysokým stupn m
zabezpe ení

- “ je objektov orientovaný, umo0uje vzak i klasické procedurální programování
- “ vytvo ené programy jsou zcela portabilní (program vytvo ený pod MS Windows bez problém funguje pod Unixem a naopak)
- “ syntaxe výraz a p íkaz vychází z jazyka C; p echod z Javy na C nebo C++ je tedy jednoduzzí, ne0 odjinud
- “ základní implementaci (JDK . Java Development Kit) firmy Sun lze pro prost edí Windows i Unix stáhnout ze stránek firmy Sun:
<http://java.sun.com>
- “ My pou0íváme vývojové prost edí NetBeans 7., fy.
<http://www.netbeans.org/>
 - . V sou asnosti je aktuální Netbeans 7.3.1 s Javou 7.0 u25
 - . Studenti si tedy mohou snadno vývojový nástroj instalovat na svých domácích počíta ích a mohou se u it programovat i mimo počíta ové u ebny zkoly

AVA . první program

“ P íklad: program vypíše daný text na obrazovku:

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

“ Po p ekladu 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 (název programu bude tedy uložen v souboru sPrvniProgram.java)

“ **deklarace veřejné třídy (public class),**

“ **hlavní funkce main (veřejná statická metoda, public static method)**

“ Hlavička funkce funkce main ():

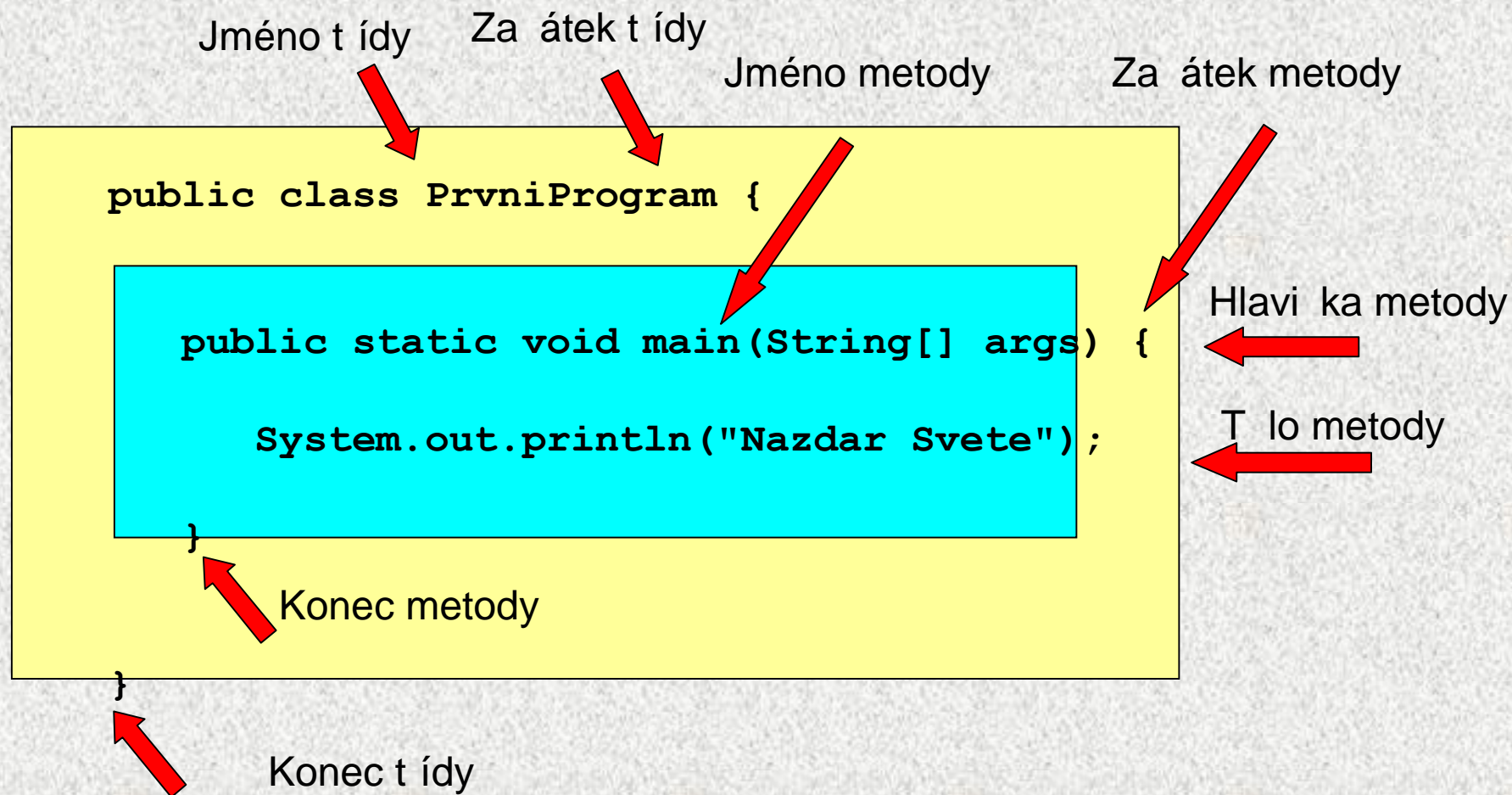
“ klíčová slova **public static void** (void - procedura)

• **(String[] args)** specifikace vstupních parametrů

“ Konvence: jména třídy se píšou 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)



Algoritmus, slovní popis, analýza

- “ Úloha:
Najdte největšího společného dělitele čísel 6 a 15

- “ řešení:
Popíšeme postup tak, aby byl použitelný pro dvě libovolná přirozená čísla, nejen pro 6 a 15:
 - “ označme zadaná čísla x a y a mezi 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

ritmus . spr b h výpo tu%

Úloha: najd te nejv tžího spole ného d litele ísel 6 a 15

Pr b h ezení:

krok	x	y	d	poznámka
	6	15	?	zadání vstupních dat
1	6	15	6	
2	6	15	6	d není d litelem y , prove krok zmenzení d
3	6	15	5	
2	6	15	5	d není d litelem x , prove krok zmenzení d
3	6	15	4	
2	6	15	4	d není d litelem x ani y , prove krok zmenzení d
3	6	15	3	
2	6	15	3	d je d litelem x i y , prove krok 4
4	6	15	3	výsledek je hodnota 3

Algoritmus - největší společný dělitel

- “ Úloha:
najdte největší společný dělitel

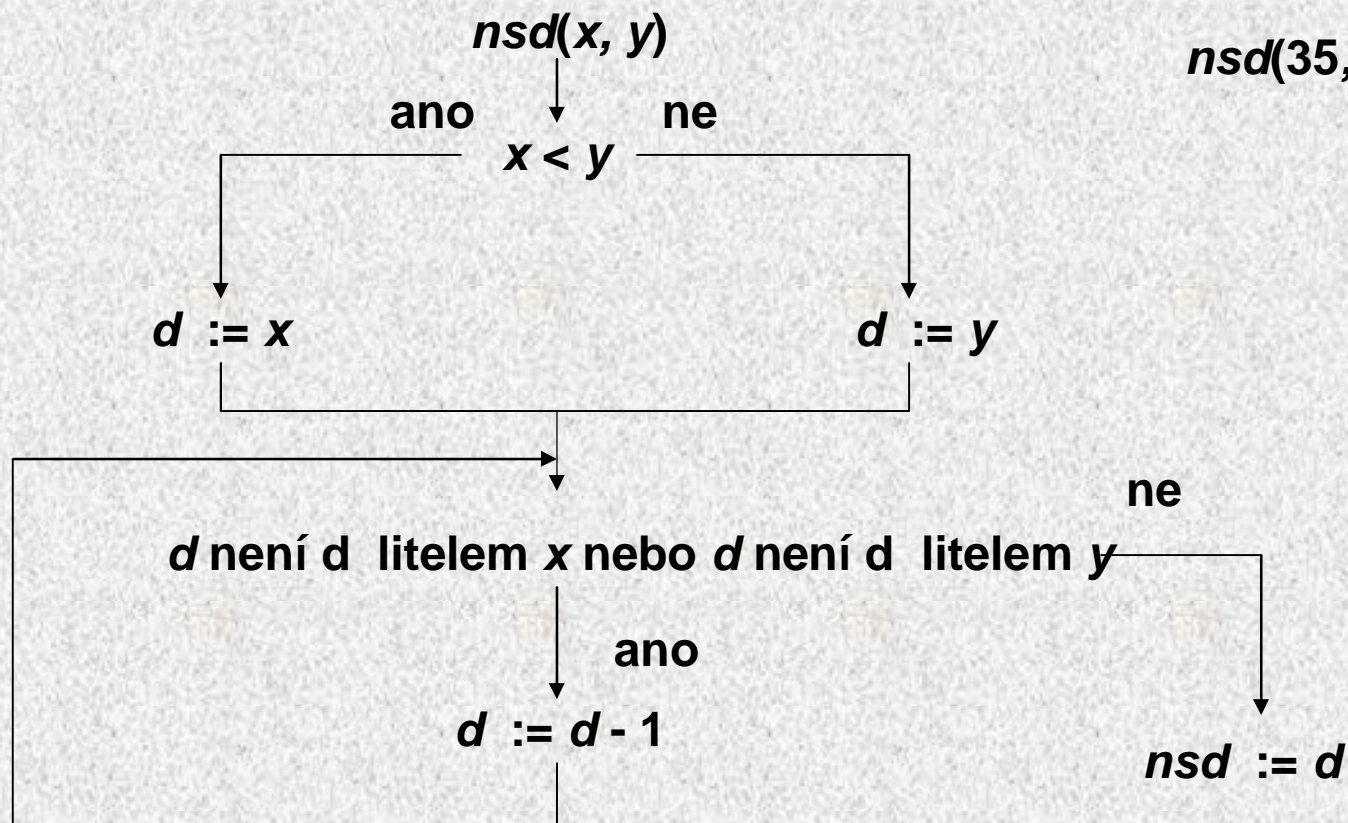
- “ Obecný popis:
Vstup: pairozená čísla x a y
Výstup: $nsd(x,y)$
Postup:
 1. Je-li $x < y$, pak d má hodnotu x , jinak d má hodnotu y
 2. Opakuj krok 3, pokud d není dělitelem x nebo d není dělitelem y
 3. Zmenš d o 1
 4. Výsledkem je hodnota d

- “ Sestavili jsme algoritmus pro výpočet největšího společného dělitele dvou pairozených čísel

Algoritmy, vývojový diagram

” Vývojový diagram

Výpočet pro
 $nsd(35, 7)$



Algoritmus

“ Algoritmus

- postup p í ezení ur íté t ídy úloh, který je tvo en seznamem **jednozna n definovaných p íkaz** a zaru uje, že pro **každou p ípustnou kombinaci vstupních dat** se po provedení **kone ného po tu krok** dosp je k **pořadovaným výsledk m**

“ Vlastnosti algoritmu:

- ***hromadnost***
m nitelná vstupní data
- ***determinovanost***
každý krok je jednozna n definován
- ***kone nost a resultativnost***
pro p ípustná vstupní data se po provedení kone ného po tu krok dojde k pořadovaným výsledk m

“ Algoritmus . syntetický model postupu ezení obecných úloh

“ Prost edky pro zápis algoritmu

- p írozený jazyk, vývojové diagramy, struktogramy, pseudojazyk, programovací jazyk

Algoritmy a programování

“ 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;
}
  
```

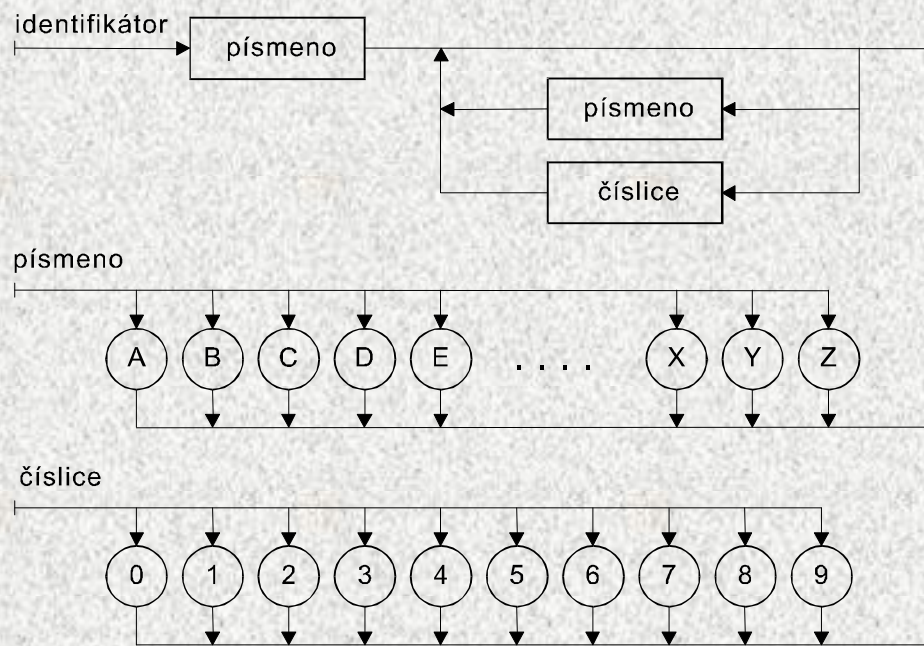
d--; příkaz, který sníží hodnotu uloženou v proměnné d o jedničku

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

Pravidla programovacích jazyk

” 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'

čísllice = '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '9'

“ Neterminály:

identifikátor, písmeno, číslice

“ Terminály:

$\neq Aq \neq Bq \dots$

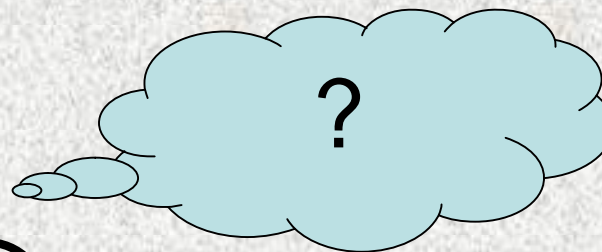
“ Význam metasymbol :

$\{x\}$ 0-ádný nebo n-á kolik výskyt x

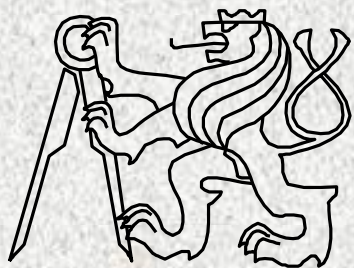
$x | y$ x nebo y

$[x]$ 0-ádný nebo jeden výskyt x

Vlastní studium



Programování 1 . úvod KONEC



A0B36PR1-Programování 1
Fakulta elektrotechnická
eské vysoké učení technické