Slajdy k přednáškám předmětu

# POČÍTAČE A PROGRAMOVÁNÍ 1

http://portal.zcu.cz

KIV/PPA1

Arnoštka Netrvalová



Slidy jsou určeny studentům předmětu Počítače a programování 1 vyučovaného katedrou informatiky a výpočetní techniky fakulty aplikovaných věd Západočeské univerzity v Plzni jako doplněk a podpora přednášek. Jakékoliv jiné využití těchto slidů podléhá souhlasu autorky.

Základním studijním materiálem je záznam přednášek z předmětu Počítače a programování 1 - 2009 vytvořený Pavlem Heroutem, jemuž za možnost využití materiálu tímto děkuji.

*A. Netrvalová*

# Obsah 1. přednášky:

- O předmětu
- Trocha počítačové historie
- Počítač dnes
- Problém – algoritmus – program
- Java – editace, překlad, spuštění
- Komentáře a dokumentace kódu
- Grafický nástroj DrawingTool

Tato tematika je zpracována v

**Záznamy přednášek:** str. 1-13, 31-33

# O předmětu (podrobně portal.zcu.cz – nutno přečíst)

## 1. Požadavky

### Absolvování + zisk bodů

- domácí úlohy (DÚ)
- písemný test (T)
- samostatná práce (SP)
- písemná zkouška (P)

### Něco se nepovedlo?

- stanoven min. a max. počet bodů z každé kategorie (DÚ, T, SP, on-line test ONL, P).
- možnost opravy
- bonusy (ACM úlohy, vybrané DÚ, SP rozšíření)

**Výsledná známka (součet bodů za celý semestr)**

## 2. Kritéria hodnocení

| Kategorie | max | min |
|---|---|---|
| DÚ | 18 | 9 |
| T | 25 | 7 |
| SP | 7 | 5 |
| ONL | 10 | 0 |
| P | 40 | 16 |
| Σ | 100 | 37 |

| Bonusy | navíc |
|---|---|
| DÚ | 6 |
| SP | 4 |
| ACM | 10 |
| Σ | 20 |

| Známka | Σbodů |
|---|---|
| 1 | ≥90 |
| 2 | ≥70 |
| 3 | ≥50 |
| 4 | <50 |
| Σmin | 37!!! |

# 3. Etika, opisování a práce na zakázku

**Poradit není totéž, co opsat!**

**DÚ, - kontrola shodnosti validovaných řešení**
**SP, ACM - obhajoba**

**T, P, ONL - kontrola identity, dozor**

**Kontrola počtu získaných bodů:**
**práce „pod dohledem“ vs. práce ostatní.**

## Definition of Academic Dishonesty

Academic dishonesty is any action or attempted action that may result in creating an unfair academic advantage for yourself or an unfair academic advantage or disadvantage for any other member or members of the academic community.

University of California, Berkeley

Více o etice: portal.zcu.cz – Studijní materiály

# Trocha počítačové historie ...

3000 př.n.l Abacus - mechanické sčítání a odčítání

1645 mechanická sčítačka Pascaline (Blaise Pascal)

1830 Charles Babbage – návrh Difference Machine

1890 vynález děrného štítku Punched Cards

1939 vynález elektronkového počítacího stroje

1943 Bletchley Park - stroj k prolamování šifer

1946 1. počítač ENIAC (elektronkový, 1. generace)

1959 počítač (transistorový, 2. generace, miniaturizace)

1965 počítač (integrovaný obvod, 3. generace), NASA - program dobývání Měsíce

1975 první počítač pro domácnost - Altair 8800 (kit), TV, mg kazeta

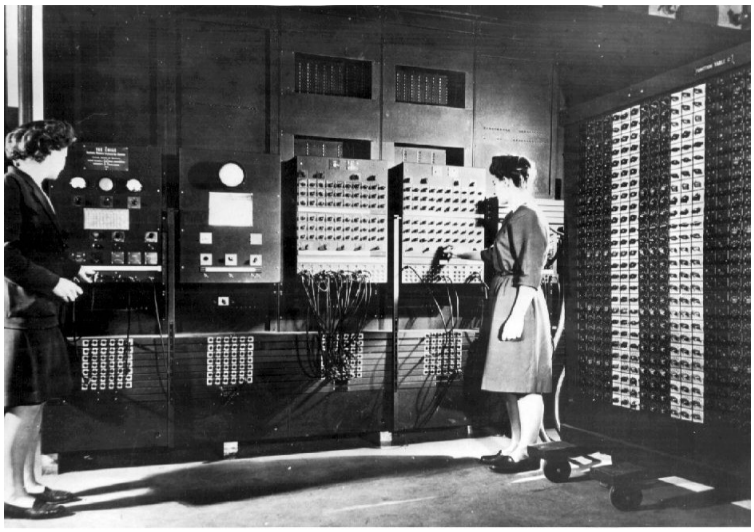
1977 Apple II – assembler

1981 IBM PC

1992 notebook

Společnost IBM PCD ThinkPad

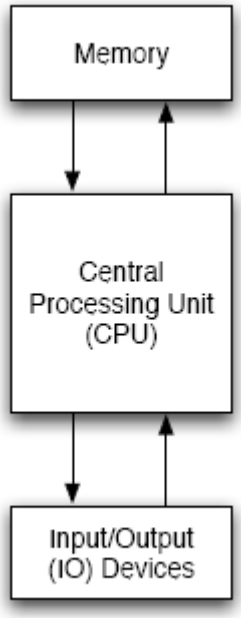
...

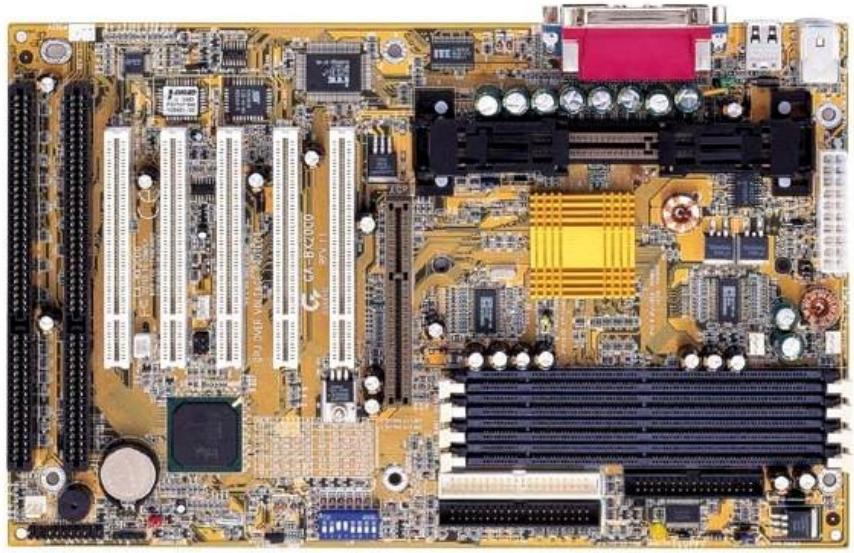


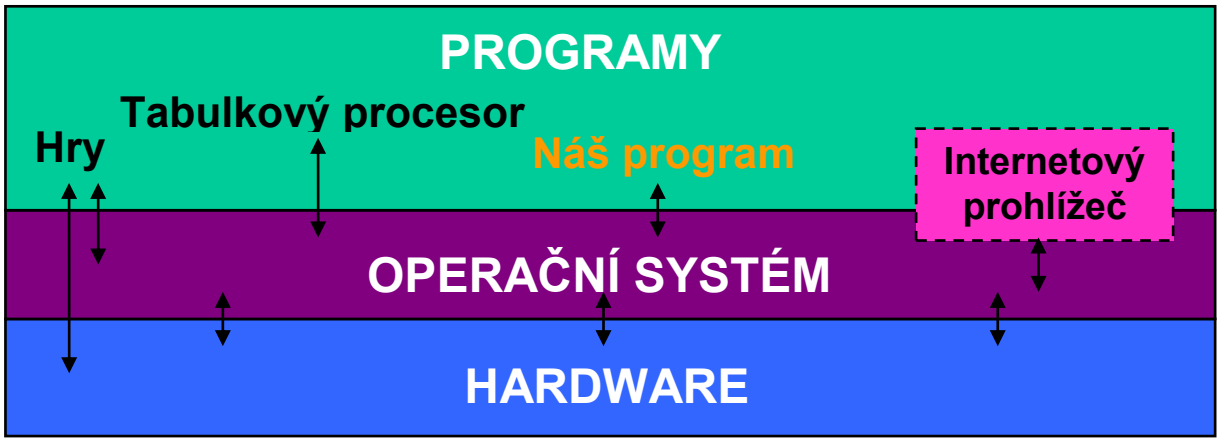
Zdroj: http://www.lancs.ac.uk/phy281

# Počítač „dnes“

Obrázky: http://www.lancs.ac.uk/phy281





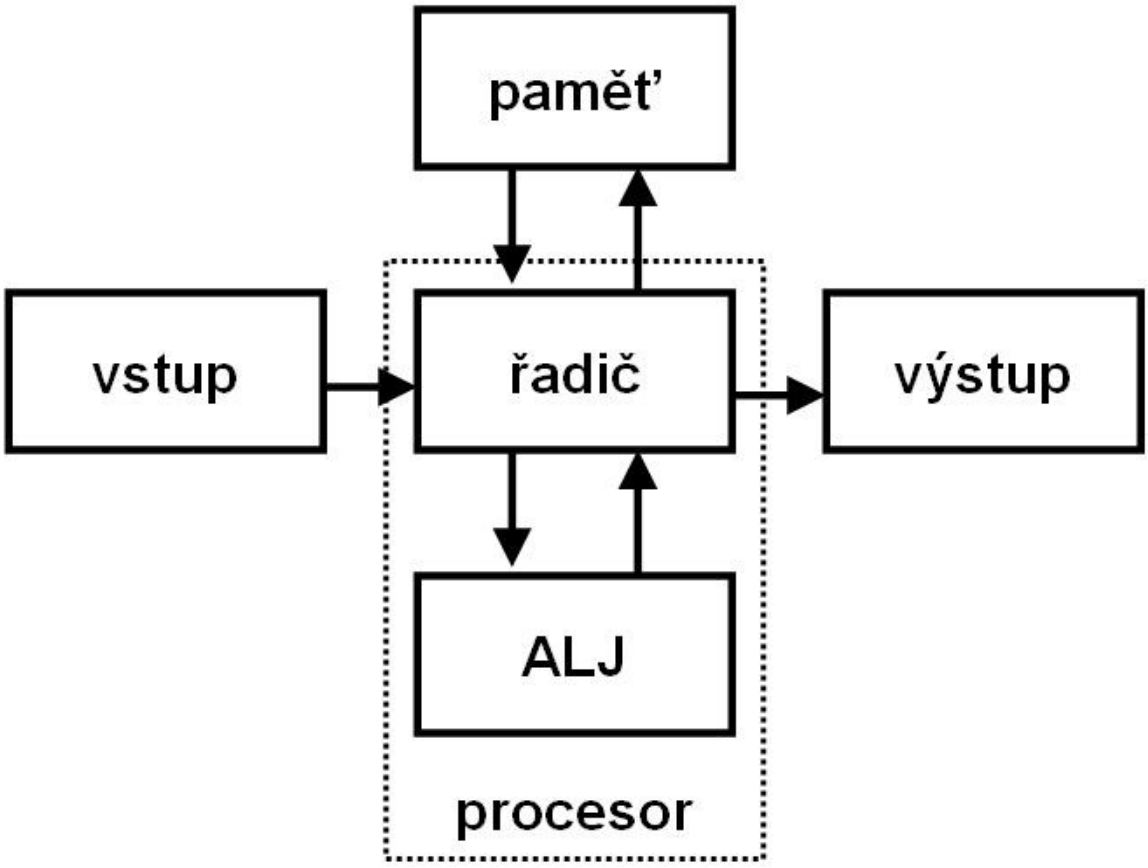

**Paměti** (vnitřní, vnější)

1bit = 0 nebo 1, 8bitů = byte [B]
1KB = 1024 bytů, 1MB = 1024 KB, 1GB =1024 MB, ...

**Pozor: 1024 [K] není 1000 [k]**
Př.: 19,5 KB = 19,5 x 1024 = 19968 Bytů

## Struktura počítače z pohledu programátora

- koncepce von Neumanna




Vstup - klávesnice, soubor na HD, myš, scanner, mikrofon, CD, ...
Výstup – obrazovka, soubor na HD, plotter, reproduktory, vypalovačka,...

Vstup + výstup = periferní zařízení (I/O)

Paměť – vnitřní RAM
Řadič (řídicí jednotka controller)
ALJ – aritmeticko-logická jednotka

# Problém – algoritmus – program

## 1. Problém

Definice (Slovník spisovného jazyka českého)
"*Věc k řešení, nerozřešená sporná otázka, otázka k rozhodnutí, nesnadná věc*"

Definice (informatika):
Každý problém P je určen uspořádanou trojicí (**vst; vyst; fce**), kde **vst** představuje množinu přípustných vstupů, **vyst** množinu očekávaných výstupů, **fce: vst→vyst** funkci přiřazující každému vstupu očekávaný výstup.

Příklad problému
... nalézt cestu do hotelu, příprava těstovin, dostat se do autobusu č. 30, 2x výpis Ahoj, ...

## 2. Algoritmus

Pojem cca 1000 let starý - významný perský matematik Abú Abdallah Muhammad ibn Musa al-Khwarizmi

**Algoritmus**

**= obecný předpis sloužící pro řešení zadaného problému**

- posloupnost kroků doplněná jednoznačnými pravidly

Vlastnosti:

- determinovanost
- rezultativnost
- hromadnost
- opakovatelnost
- efektivnost

Přístupy k návrhu algoritmu:

dekompozice a abstrakce problému
metodologie „shora dolů“ a „zdola nahoru“.

Prostředky pro zápis algoritmů:

- vývojový diagram
- pseudojazyk
- programovací jazyk

Příklad algoritmu
- vypiš slovo Ahoj dvakrát

# 3. Program

- zápis algoritmu některým z uvedených prostředků

**Algoritmus + data = program**

## Programovací jazyk

syntaxe (pravidla) a sémantika (význam)
strojově orientované vs. vyšší programovací jazyky
Metody implementace: interpretační, kompilační

## Interpret

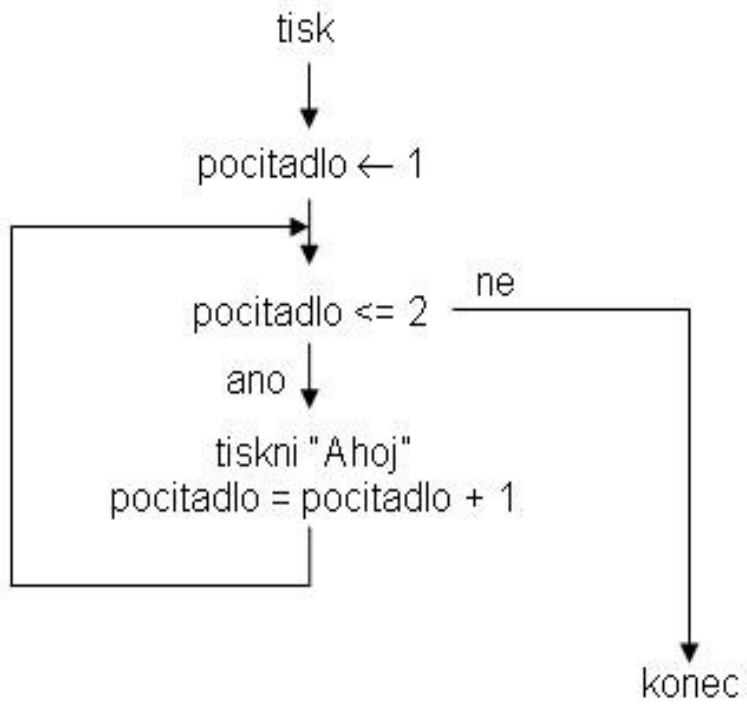
editace→zdrojový kód+vst. data → interpret→ výst.data

## Kompilátor

editace→zdrojový kód→kompilátor→
→cílový program + vst.data→výstupní data

## Programovací jazyky

Algol, Simula, Cobol, Fortran, Basic, Pascal, Delphy, C, C++, C#, LISP, Java, Python, Prolog, ADA, SQL,...



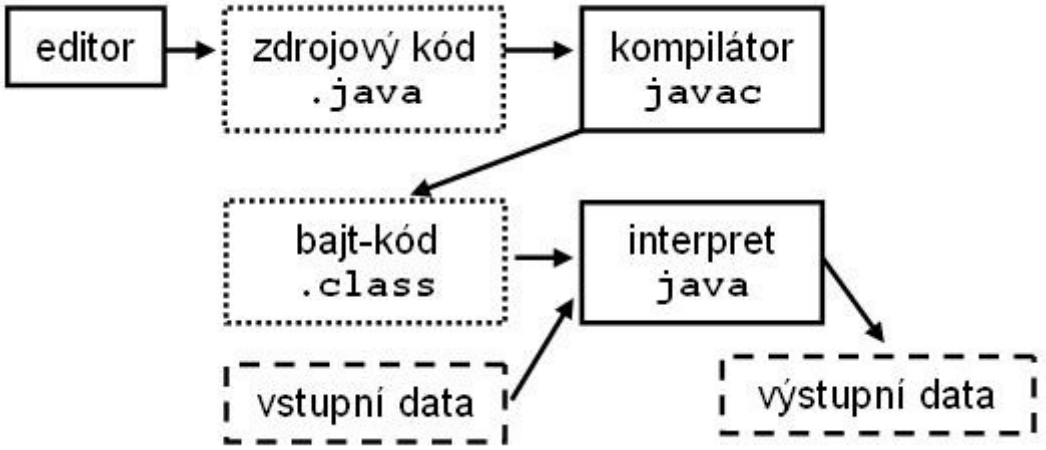

```
pocitadlo = 1
dokud je pocitadlo ≤2
begin
   println(Ahoj)
   zvys pocitadlo o 1
end
```

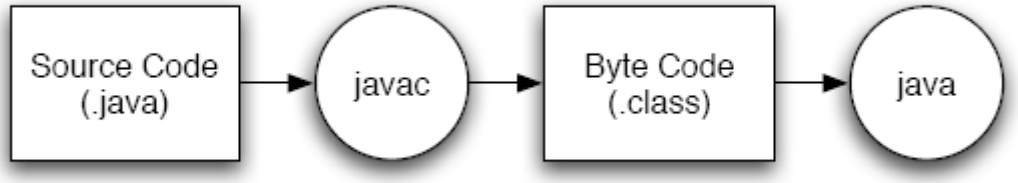
```
public class Ahoj {

  public static void main(String[] args) {
     int pocitadlo = 1;
      while (pocitadlo <=2) {
          System.out.println("Ahoj");
          pocitadlo = pocitadlo + 1;
     }
  }
}
```

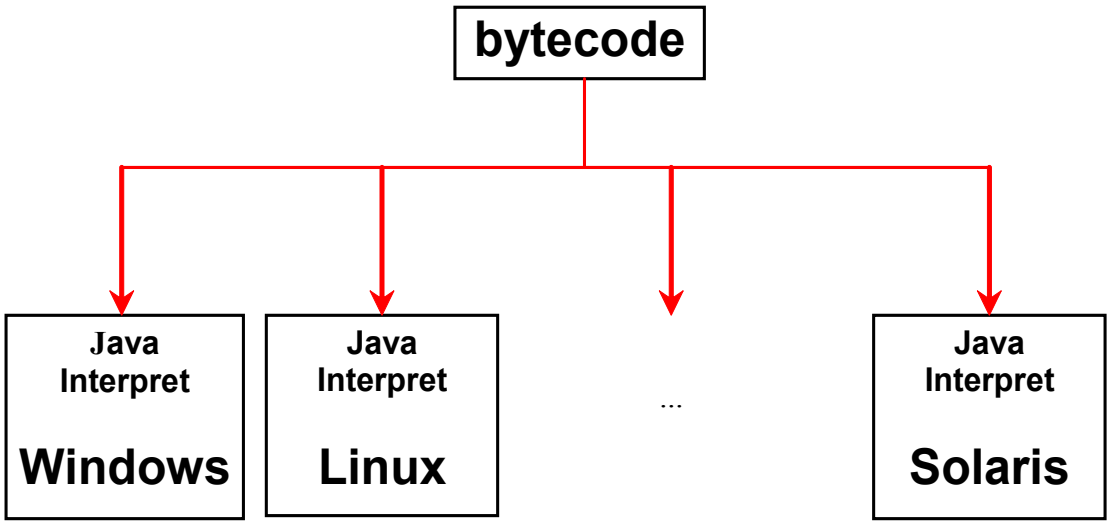
```
Ahoj
Ahoj
```

# Java – editace, překlad, spuštění




nezávislost na platformě



## Java

- James Gosling, Oak
- Java, 1995, Sun
- 2004 – JDK 1.5 – nové prvky jazyka
- 2006 – JDK 1.6 – verze aktuální pro PPA1
- 2011 – JDK 1.7 – nová verze

## Program

- jeden či více zdrojových souborů **Jmeno.java**
- překlad do vnitřní formy (bytekód)
- vznikne soubor **Jmeno.class**
- interpretaci provádí program java JVM (Java Virtual Machine)
- program využívá knihovny (Java Core API - Application Programming Interface)
- JVM + Java Core API = Java platforma
- program v Javě = třída(y)

## Vývojové prostředky

- JDK+ příkazová řádka
- SciTe
- Eclipse

## Instalace Javy

- dle návodů na portálu (vše ke stažení)
- nejprve instalace Javy pak Scite a Eclipse

Další informace:

http://java.sun.com
http://www.scintilla.org
http://www.eclipse.org/platform

# První program v Javě

## Zásady:

- program (zdrojový soubor) vytváříme v textovém editoru (Scite, NotePad, WordPad, PSPad), Eclipse
- název třídy začíná velkým písmenem, ve více slovném názvu začíná každé slovo velkým písmem, nepoužívá se podtržítko ani pomlčka, v názvu nesmí být mezera, třída tvoří blok **{ . . . }**
- ukládaný zdrojový soubor musí mít název shodný s názvem třídy a to včetně malých a velkých písmen
- **uložení** zdrojového souboru s příponou *.java
- **překlad: javac** Pozdrav.java
- **spuštění: java** Pozdrav

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

## Poznámka (bude vysvětlováno i později):

modifikátor **public** – přístupová práva k třídě, metodě, atributu
Metoda **public static void** main(String[] args) je veřejná statická metoda, jejíž hlavička musí být zapsána shodně s výše uvedeným příkladem!
Parametrem metody (metoda má název s počátečním malým písmenem, vyjadřující děj) je pole řetězců pojmenované args (může být i jinak) představující argumenty (možnost zadání vstupních dat z příkazové řádky, např. jméno souboru).
Metoda main() nevrací žádnou hodnotu, její návratový typ je vždy(!) **void**.

## Druhý program

```
public class Ahoj {
    public static void main(String[] args) {
        int pocitadlo = 1;
        while(pocitadlo <=2) {
            System.out.println("Ahoj");
            pocitadlo = pocitadlo + 1;
        }
    }
}
```

Poznámka:

Příkaz System.out.println("Ahoj"); je voláním metody pro výpis na obrazovku, metoda pracuje se standardním výstupem (out), patří do třídy System, která je připojena automaticky (není třeba import).

Výraz "Ahoj" představuje řetězec, což je text uzavřený v řetězových závorkách " " (toto je prázdný řetězec).

## Java a čeština

```
public class PozdravČesky {
  public static void main(String[] args) {
    System.out.println("Ahoj světe!");
    System.out.println("Žluťoučký kůň příšerně úpěl ďábelské ódy.");
  }
}
```

Java češtinu „umí“, výstup je v implicitním kódování operačního systému, proto je nutno předchozí program upravit a sadu nastavit, jinak nebude „všude“ správně fungovat!

**Češtinu nepoužívat ani v programech ani v komentářích!**

# Komentáře a dokumentace

- snazší orientace ve zdrojovém kódu

Typy
- řádkový (do konce řádku) // text
- blokový (více řádek) /* text */
- dokumentační /** dokumentace */

**Co dokumentovat?**
- nestandardní názvy proměnných
- nestandardní programové konstrukce
- metody
- třídy

**Vygenerování dokumentace**
- program **javadoc**
- příkaz javadoc –d mujDokument Jmeno.java
- založí adresář mujDokument s html soubory
- spustit Index.html

Jméno autora (v dokumentaci - **@author**) a příkaz:
javadoc –author –d mujDokument Jmeno.java

Příkaz pro dokumentování více souborů:
javadoc –author –d mujDokument *.java

A pokud není užit modifikátor public:
javadoc –author – private –d mujDokument *.java

## Příklad dokumentace

- pro již uvedený program Ahoj.java

```
import java.io.*;
// zde import netřeba, jen pro ukazku - kam dokumentaci umistit

/**
 * Dva vypisy slova Ahoj
 * @author netrvalo
 * @version 1 - 26.07.2010
*/
public class Ahoj {
/**
 * Metoda main() - spousteni programu
 * @param args pole retezcu argumentu
 * @return nevraci zadnou hodnotu
*/
  public static void main(String[] args) {
    int pocitadlo = 1;   // reprezentuje pocet vypisu
  /*
     cislo 2 muzeme zamenit za 3
     pak se vypis provede 3x
  */
    while(pocitadlo <=2) {
      System.out.println("Ahoj");
      pocitadlo = pocitadlo + 1;
    }
  }
}
```

# Grafický nástroj DrawingTool[1]

- vytvořen pro PPA2 pro kreslení fraktálů
- umí nakreslit čáru z bodu do bodu
- počátek grafického souřadného systému 0,0 je v levém horním rohu (pracuje se s pixely)
- soubor DrawingTool.java musí být uložen ve stejném adresáři s naší aplikací (DrawingTool je v podstatě samostatná třída, kterou náš program používá)
- aplikace musí mít na 1. řádku příkaz: **import**.java.awt.*;

**Příklad použití – na cvičení**

Soubor DrawingTool.java a aplikace Kreslení .java

překlad: javac *.java // prelozi obe tridy

spuštění: java Kresleni

**Problém:**

Nakreslete úhlopříčku vedoucí z levého horního do pravého dolního rohu černou barvou na bílém pozadí v okně s rozměry 300x200.

[1] © 2005 Ing. Roman Tesař (příklady studenti doktorského studia na KIV), pro předmět PPA2.

```
import java.awt.*;
public class Kresleni {
  public static void main(String[] args) {
    /* inicializace: sirka = 300,
       vyska = 200, barva pozadi = bila
    */
    DrawingTool dt =
         new DrawingTool(300, 200, Color.WHITE, true);
    // nastaveni barvy cary na cernou
    dt.setColor(Color.BLACK);
    /* nakresleni uhlopricky z leveho horniho rohu [0, 0]
       do praveho dolniho rohu [300 - 1, 200 - 1]
    */
    dt.line(0, 0, 299, 199);
  }
}
```

Výstup: