

# Algoritmizace a programování

## Struktura programu

- Vytvoření nové aplikace
- Struktura programu
- Základní syntaktické elementy
- První aplikace

# Nová aplikace – NetBeans

- Ve vývojovém prostředí NetBeans
  - File
  - New Project
  - Java Application
  - Project Location
  - Project Name
  - Create main class

```
package pkg_nic_moc;  
  
/**  
 *  
 * @author Jirina  
 */  
public class Main {  
  
    /**  
     * @param args the command  
     * line arguments  
     */  
    public static void main(String[] args) {  
        // TODO code application here  
    }  
}
```

# Nová aplikace – Eclipse

- Při spouštění Eclipse nutno zadat Workspace tj. umístění projektů
- Ve vývojovém prostředí Eclipse
  - File
  - New
  - Java Project
- V aplikaci, src
  - New
  - Package
- V package
  - New
  - Class

```
package pkg_nic_moc;

public class Nic {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub

    }
}
```

# První pokus

```
package pkg_nic1;

public class Prvni {

    /**
     * @param args
     */
    public static void main(String[] args) {
        /* vypis na konzoli */
        System.out.println("Muj prvni program!");
        // konec programu
    }
}
```

# Struktura programu

- Jazyk Jáva je objektový – obecné šablony objektů jsou třídy (class)
- Program sestává minimálně z jedné, zpravidla z více tříd.
- Každá třída musí být uložena v souboru stejného jména jako třída, zdrojový soubor každé třídy má příponu java (název třídy a název souboru jsou totožné).
- Třídy jsou členěny do balíčků (package).
- Struktura balíčků (struktura pojmenování) musí odpovídat struktuře adresářů.
- Zařazení do balíčků znamená mimo jiné umístění zdrojového souboru do příslušného adresáře.
- Vstupním bodem programu může být pouze třída obsahující metodu main deklarovanou jako

```
public static void main (String[] args) { // prikazy}
```

# Lexikální struktura, lexikální (syntaktické) elementy

- Program v Javě – znaková sada Unicode – abeceda jazyka obsahuje všechny znaky Unicode
- **Prázdná místa** – mezery, tabulátory, znaky konce řádku
- **Komentáře**
  - Jednořádkový //
  - Víceřádkový /\*            \*/
  - Dokumentační /\*\*            \*/
- **Klíčová slova**
- **Identifikátory**
  - Způsob zápisu identifikátorů
  - Case sensitive
- **Literály** – konstantní hodnoty
- **Separátory, oddělovače**
- **Operátory**
  
- Jazyk Java je „**case sensitive**“ – při zápisu kódu jsou rozlišována malá a velká písmena

# Klíčová slova jazyka Java

- Rezervovaná slova – nelze je používat jako identifikátory
  - **Klíčová slova**
  - Konstrukce programu – zápis příkazů a podobně

```
abstract  assert  boolean  do  break  
byte  case  catch  char  class  
const  continue  default  double  else  
enum  extends  final  finally  
float  for  goto  if  implements  
import  instanceof  int  interface  
long  native  new  package  
private  protected  public  return  
short  static  strictfp  super  switch  
synchronized  this  throw  throws  
transient  try  void  volatile  while
```

- **Další** rezervovaná slova
- Literály typu boolean
- Literál null

```
true, false  
null
```

# Identifikátory v jazyce Java

- Identifikátor = jméno
  - Třídy, rozhraní
  - Proměnné, datové položky tříd, členské proměnné tříd
  - Metody tříd
  - Konstanty
  - Balíčky
- Identifikátor
  - Sekvence písmen, číslic, znak podtržítka, nesmí začínat číslicí
    - (JavaLetter, JavaDigit)
  - Pro identifikátory nelze používat vyhrazená slova
  - Délka identifikátoru není omezena
  - Znak tečka „.“ odděluje jednotlivé části ve složených identifikátorech
- Doporučení
  - **Třídy a rozhraní** – identifikátor začínat velkým písmenem, další slova v identifikátoru začínat velkým písmenem  
`String, StringBuffer`
  - **Proměnné a metody** – identifikátor začínat malým písmenem  
`pocetCisel, getSize()`
  - **Konstanty** – používají se pouze velká písmena, ve víceslovných identifikátorech se používá znak podtržítka „\_“ pro oddělení slov  
`MAX_HODNOTA`
  - **Balíčky** – identifikátory se skládají pouze z malých písmen, ve složených jménech je oddělovačem jednotlivých částí je znak „tečka“  
`mypkg, java.lang`

# Speciální symboly

## ■ Separátory

(            )  
{            }  
[            ]  
;            ,            .

## ■ Operátory

=	>	<	!	~
?	:			
==	<=	>=	!=	
&&		++	--	
+	-	*	/	
&		^	%	
<<	>>	>>>		
+=	-=	*=	/=	
&=	=	^=	%=	
<<=	>>=	>>>=		

# Komentáře, anotace

## ■ **Komentáře** – přehledný popis kódu

- Jednořádkový komentář  
`// toto je komentar`
- Víceřádkový komentář  
`/* toto je  
taky komentar  
*/`

Komentáře nemohou být vnořené

## □ Dokumentační komentář

```
/** toto je dokumentacni  
komentar  
*/
```

- Dokumentační komentáře mohou obsahovat značky začínající znakem @
- Jazyk Java obsahuje nástroj, který umožňuje automatické generování dokumentace programu z dokumentačních komentářů
- Pro formátování dokumentace lze využít HTML

## ■ **Anotace** – značky vkládané do zdrojového kódu, začínají znakem @

```
@Override  
@Deprecated
```

# Hlavní program

- Hlavní program – metoda, která je vyvolána po spuštění programu se musí jmenovat `main`
- Tato metoda musí být uvnitř veřejné třídy – deklarována jako `public class`
- Tato třída se musí jmenovat stejně jako soubor, ve kterém je uložena – pravidlo platí pro všechny veřejné třídy
- Metoda `main`
  - Ve třídě, která je označena `public`
  - Musí mít přesně podobu

```
public static void main(String[] args) {  
    // kod  
    // prikazy  
}
```
- Po spuštění programu se začne postupně vykonávat kód umístěný v metodě `main`

# Přeložení a spuštění programu

- Překlad programu
- Spuštění ve vývojovém prostředí
- Spuštění programu mimo vývojové prostředí – z terminálového okna (okno příkazového řádku)
  - Spuštění `class` souboru
  - Spuštění `jar` souboru
- Vytvoření, překlad a spuštění programu mimo vizuální vývojové prostředí – pomocí prostředků JDK
  - V textové editoru zapsat kód třeba v Notepad – musí se jednat o prostý textový editor, který neuchovává formátování textu a podobně, nelze použít třeba Word, vývojová prostředí ve svých editorech rovněž běžně formátují text, ovšem toto formátování není uloženo do souboru současně s editovaným textem
  - Překlad z příkazové řádky pomocí `javac`
  - Spuštění z příkazové řádky pomocí `java`

## Druhý pokus

```
package pkg_nic2;

public class Druhy {

    /**
     * @param args
     */
    public static void main(String[] args) {

        float a = 3.6f;
        float obvod = 4 * a;
        float obsah = a * a;
        System.out.println("Strana ctverce " + a);
        System.out.println("Obvod ctverce " + obvod);
        System.out.println("Plocha ctverce " + obsah);
    }
}
```

# Třetí pokus

```
package pkg_nic3;

import java.util.Scanner;

public class Treti {

    /**
     * @param args
     */
    public static void main(String[] args) {

        float a;
        Scanner sc = new Scanner(System.in);
        System.out.println("Zadej stranu ctverce ");
        a = sc.nextFloat();
        float obvod = 4 * a;
        float obsah = a * a;
        System.out.println("Strana ctverce " + a);
        System.out.println("Obvod ctverce " + obvod);
        System.out.println("Plocha ctverce " + obsah);
    }
}
```

# Shrnutí

- Vytvoření aplikace
  - Struktura programu
  - Překlad aplikace
  - Spuštění aplikace
- 
- Integrovaná vývojová prostředí – integrace nástrojů pro vývoj aplikací – editor, překladač, debugger, spuštění programu ...
- 
- Použití prostředků JDK

# Praktické ukázky

- Spuštění vývojového prostředí NetBeans, vytvoření nové aplikace, překlad, spuštění. Soubory na disku
- Spuštění vývojového prostředí Eclipse, vytvoření nové aplikace, překlad, spuštění. Soubory na disku
- První pokus – první aplikace
- Druhý pokus – program pro výpočet hodnoty výrazu
- Třetí pokus – program komunikující s uživatelem

# Úlohy k procvičování

- Spust'te vývojové prostředí NetBeans. Vytvořte novou aplikaci, přeložte, spust'te. Zjistěte strukturu menu, nabízené možnosti, klávesové zkratky základních operací. Po spuštění aplikace využívající terminálový vstup nebo výstup není vytvořeno konzolové okno – prostředí konzolu pouze emuluje – příslušná interakce je v části okna vývojového prostředí. Zkuste vyvolat kontextovou nápovědu stiskem <SHIFT>+<F1>.
- Stejným způsobem vyzkoušejte prostředí Eclipse.