

Java tahák

Kurz: Java - úvod do programování (JAV1)

Datové typy

Typ	Obsahuje	Příklady
int	Celá čísla	1, 0, -20
double	Reálná čísla	1.3, 2.0
boolean	Pravdivost	true, false
char	Znak	'a', '1', '@'
String	Text	"Jan Novák"
pole	typ [počet]	int[10], String[20]

Větvení

```
if ( podmínka ) {
    příkazy pokud podmínka je true
} else {
    příkazy pokud podmínka je false
}
```

Cyklus s podmínkou

```
while ( podmínka ) {
    příkazy
}
```

Cyklus pro pole

```
for ( Typ proměnná : pole ) {
    příkazy s proměnnou
}
```

Výstup na obrazovku

```
System.out.println( proměnná/výraz )
```

```
System.out.println() - prázdný řádek
```

Operace

Operace		Příklad
Plus, mínus	+, -	1 + 1, 3 - 2
Násobení	*	2 * 3
Dělení	/	5 / 2
Zbytek po dělení	%	5 % 2
Menší, Větší	<, >	1 < 2, 5 > 2
Je rovno	==	1 == 1
Není rovno	!=	5 != 3
Zřetězení textu	+	"Ahoj" + "!"
Logické A	&&	5 > 1 && 5 > 2
Logické ANEBO		a > 0 a < 0
Negace	!	!(5 < 2)
Přiřazení	=	int a = 10

Pole

- `typ[]` proměnná
- Pole bez položek:
 - `int[] cisla = new int[10]`
- Pole s položkami:
 - `int[] cisla = new int[] { 1, 2, 3 }`
- první prvek: **a[0]**
- počet prvků: **a.length** (bez závorek)

Zdrojové soubory

- přípona .java
- v jednom souboru jedna třída / rozhraní
- jméno souboru je jméno třídy / rozhraní

Třída (class)

Předpis pro vytvoření objektu.

```
class Název {
    proměnné
    metody
}
```

Potomek dědí od předka:

```
class Potomek extends Předek {
    proměnné
    metody
}
```

Abstraktní předek:

```
abstract class Předek {
    proměnné
    metody
    abstraktní metody (bez příkazů)
}
```

Rozhraní (interface):

```
interface R1 {
    abstraktní metody (bez příkazů)
}
```

```
class Implementace implements R1, R2 {
    proměnné
    metody ze všech rozhraní
    další metody
}
```

Viditelnost

private	viditelné uvnitř třídy
protected	uvnitř třídy a potomků
public	viditelné všude

String

<https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/lang/String.html>

Metoda	Popis
length	počet znaků
startsWith	jestli začíná textem
endsWith	jestli končí textem
substring	vyber část textu
toUpperCase	všechno na velká písmena
toLowerCase	změň na malá písmena
indexOf	najdi text uvnitř
equals	porovná texty podle obsahu
compareTo	porovná abecedně

Namísto == pro porovnání použij **equals()**

Namísto > a < použij **compareTo()**:

- `a.compareTo(b) < 0` → `a < b`
- `a.compareTo(b) == 0` → `a == b`
- `a.compareTo(b) > 0` → `a > b`

Objekty

Základní typ objektu: **Object**

<https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/lang/Object.html>

- implicitní předek všech objektů

Vytvoření objektu:

```
Object o = new Object();
```

Vytvoření s parametry:

```
MujObjekt o =
    new MujObjekt(param1, param2);
```

Přiřazení do proměnné předka:

```
Predek o = new Potomek();
```

Přiřazení do proměnné rozhraní:

```
Rozhání o = new Implementace();
```

Balíky (packages)

Balíky zeskupují třídy / rozhraní k sobě.
Balík je něco jako adresa třídy / rozhraní.

```
package priklady;
```

- v adresáři **priklady**

```
package moje.priklady
```

- v adresáři **priklady**, který je uvnitř adresáře **moje**
- Např: `moje\priklady\MojeTrida.java`

Před použitím třídy/rozhraní z jiného balíku je potřeba jejich import.

Soubor **moje\priklady\Příklad1.java**:

```
package moje.priklady;
import moje.vypocty.Vypocet1;
```

```
public class Příklad1 {
    Vypocet1 vypocet;
}
```

Import všech tříd a rozhraní z balíka **moje.vypocty**:

```
import moje.vypocty.*;
```

Výjimky

```
try {
    příkazy
    (můžou být přerušeny výjimkou)
} catch (Exception e) {
    příkazy pro obsluhu výjimky
} finally {
    příkazy
    (vykonají se vždy)
}
```

Přerušeni výjimkou

```
void metoda() throws Exception {
    throw new Exception("Toto je výjimka");
}
```

Metoda bez parametrů:

```
void názevMetody() {
    příkazy
}
```

Zavolání:

```
o.názevMetody();
```

Metoda s parametry:

```
void názevMetody(Typ p1, Typ p2) {
    příkazy (p1 a p2 jako proměnné)
}
```

Zavolání:

```
String text = "Ahoj";
o.názevMetody(text);
```

Metoda s výstupem:

```
TypVystupu metoda() {
    příkazy
    return hodnota;
}
```

Zavolání:

```
String text = o.názevMetody();
```

Instanceof

Zápis:

proměnná instanceof Typ

`o instanceof B` je `true` když jedno z:

- objekt `o` je typu `B`
- objekt `o` je potomek `B`
- objekt `o` implementuje rozhraní `B`

Bezpečná změna typu

```
Predek o = new Predek();
if ( o instanceof Potomek) {
    Potomek p = (Potomek)p;
}
```

Statické metody

Nepracují s objektem a jeho proměnnými.

```
public class Vypocty {  
    public static int mocnina(int c) {  
        return c * c;  
    }  
}
```

Zavolání:

```
int vysledek = Vypocty.mocnina(5);
```

Zpuštění programu

Program začíná v statické metodě main:

```
package priklad;  
public class Aplikace {  
    public static void main(String[] args) {  
        příkazy  
    }  
}
```

Vybranou třídu lze zpustit buď přímo z IDE,
nebo z příkazového řádku:

```
java priklad.Aplikace
```