

Jaké druhy komentářů jsou v následujícím kódu?


```
/**
```

```
 * Hlavní třída programu.
```

```
 */
```

```
public class Hlavni {
```

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

```
        /* vytiskne odpověď */
```

```
        System.out.println( 42 ); // odpověď je 42
```

```
    }
```

```
}
```

:c1 jednořádková

:c2 výceřádkový

:c3 dokumentační

:c4 kompilační

:c1:c2:c3 ok ex

--

Číslo 256 457 můžeme uložit do datových typů

:c1 byte

:c2 short

:c3 int

:c4 long

:c3:c4 ok ex

--

Úkolem Java Virtual Machine je

:r1 Poskytovat programům vždy stejné prostředí

:r2 Kompilace kódu

:r3 GUI pro tvorbu programu

:r1 ok

--

Číslo 315 můžeme uložit do datových typů

:c1 byte

:c2 short

:c3 int

:c4 long

:c2:c3:c4 ok ex

--

Číslo 2,6 můžeme uložit do datových typů

:c1 int

:c2 long

:c3 float

:c4 double

:c3:c4 ok ex

--

Jakou hodnotu bude mít proměnná c

```
int a = 5;
```

```
int b = 6;
```

```
boolean c = a<=b;
```

:c1 true

:c2 false

:c3 chyba

:c1 ok ex

--

Jakou hodnotu bude mít řetězec s

```
int x = 82;
```

```
String s = "Vaše hmotnost je: " + x + " Kg";
```

:c1 "Vaše hmotnost je: " + x + "Kg"

:c2 Vaše hmotnost je: 42 Kg

:c3 42

:c4 chyba

:c2 ok ex

--

Jakou hodnotu bude mít řetězec s3?


```
String s1 = "jdk", s2 = "7.0";
```

```
String s3 = s1 + s2;
```

:c1 s1 + s2

:c2 jdk7.0

:c3 jdk

:c4 7.0

:c2 ok ex

--

Pro jednořádkový komentář používáme znaky

:c1 ++

:c2 //

:c3 --

:c4 **

:c2 ok ex

--

Co bude výstupem:

```
int m = 6;
```

```
System.out.printf( "Slon africký váží %d tun", m );
```

```
:r1 Slon africký váží %d tun
```

```
:r2 Slon africký váží 6d tun
```

```
:r3 Slon africký váží 6 tun
```

```
:r4 Chyba
```

```
:r3 ok
```

```
--
```

```
Jakou hodnotu bude mít proměnná y?<br />
```

```
int x = 1;
```

```
int y = x++;
```

```
:r1 0
```

```
:r2 1
```

```
:r3 2
```

```
:r4 3
```

```
:r2 ok
```

```
--
```

```
Hodnota a bude:<br />
```

```
int x=0;<br />
```

```
int y=5;<br />
```

```
boolean a = x == 0 && y <= 0;<br />
```

```
:r1 false
```

```
:r2 true
```

```
:r3 chyba
```

```
:r1 ok
```

```
--
```

```
Hodnota a bude:<br />
```

```
int x=0;<br />
```

```
int y=5;<br />
```

```
boolean a = x == 0 || y &lt;= 0;<br />
```

```
:r1 true
```

```
:r2 false
```

```
:r3 chyba
```

```
:r1 ok
```

```
--
```

```
Hodnota x bude<br />
```

```
<br />
```

```
int x=24;<br />
```

```
x /= 2;<br />
```

```
:r1 24
```

```
:r2 2
```

```
:r3 12
```

```
:r4 chyba
```

```
:r3 ok
```

```
--
```

```
Hodnota y bude:<br />
```

```
int x = 5;<br />
```

```
int y = x + ++x;
```

```
:r1 9
```

```
:r2 10
```

```
:r3 11
```

```
:r4 12
```

```
:r3 ok
```

```
--
```

```
Vypíše se:<br />
```

```
int mesic =3; <br />
```

```
boolean jeMaj = (mesic == 5);
```

```
if( jeMaj ) {
```

```
    System.out.println( "je lásky čas" );
```

```
}<br />
```

```
else {<br />
```

```
System.out.println( "neni maj" )<br />
```

```
}
```

```
:r1 je lásky čas
```

```
:r2 neni maj
```

```
:r3 true
```

```
:r4 false
```

```
:r2 ok
```

```
--
```

```
Tento cyklus proběhne:<br />
```

```
<br />
```

```
int x = 5;<br />
```

```
while( x &gt;= 0 ) { <br />
```

```
    System.out.println( x );<br />
```

```
    x --;<br />
```

```
}
```

```
:r1 ani jednou
```

```
:r2 2x
```

```
:r3 3x
```

```
:r4 4x
```

```
:r5 5x
```

```
:r6 6x
```

```
:r6 ok
```

```
--
```

Tento cyklus proběhne


```
int x = 5;<br />
```

```
do {<br />
```

```
System.out.println( x );
```

```
    x --;
```

```
} while ( x &lt; 0);
```

:r1 nekonečně mnohokrát

:r2 0x

:r3 1x

:r4 2x

:r5 5x

:r1 ok

--

Kolikrát proběhne tento cyklus:

```
for( int a = 1; a &lt; 4; a++ ) {
```

```
    System.out.println( a );
```

```
}
```

:r1 0x

:r2 1x

:r3 2x

:r4 3x

:r5 4x

:r6 nekonečně krát

:r4 ok

--

Kolikrát proběhne tento cyklus:

```
int s = 99;
```

```
while( s &gt; 0 ) {
```

```
    if( s%10 == 0 ) {
```

```
        break;
    }
    s = s-1;
    System.out.println( s );
}
:n
:n="9" ok
--
```

K čemu slouží dědičnost

```
:c1 koncové (final) metody rozšiřovaná třída zdědí, ale nemůže je překrýt.
:c2 Rodičovská třída dědí všechny metody mateřské třídy
:c3 Rozšiřovaná třída zdědí všechny nesoukromé metody a třídní proměnné předka
:c4 K utváření hierarchie tříd
:c1:c3:c4 ok ex
--
```

K čemu slouží metody

```
:r1 K matematickým výpočtům
:r2 Ke statistickým výpočtům
:r3 K několikerému používání stejné posloupnosti příkazů
:r4 K tvorbě objektů
:r3 ok
--
```

K čemu může sloužit return u metod

```
:c1 Okamžité ukončení metody
:c2 Vráť na začátek kódu
:c3 Vráť hodnotu
:c4 Vráť na začátek metody
:c1:c3 ok ex
--
```

Jakou hodnotu bude mít prvek pole s indexem 3?


```
int[] p = new int[10];
for( int i = 0; i < p.length; i++ ) {
    p[i] = i+2;
}
```

:c1 3

:c2 4

:c3 5

:c4 6

:c3 ok ex

--

Jakou hodnotu bude mít prvek s indexem 3?

```
int[] cisla = { 3, 5, 6, 7};
```

:c1 3

:c2 5

:c3 6

:c4 7

:c4 ok ex

--

Jakou hodnotu bude mít prvek s indexem p[2][3]

```
int[][] p = new int[4][4];
for( int i = 0; i < p.length; i++ ) {
    for( int j = 0; j < p.length; j++ ) {
        p[i][j] = j;
    }
}
```

:c1 1

:c2 2

:c3 3

:c4 4

:c3 ok ex

--

K čemu mohou sloužit třídy

:c1 K tvorbě instancí

:c2 K tvorbě objektů

:c3 Dědičnost

:c4 K definici pole

:c1:c2:c3 ok ex

--

Pokud použijem u třídy specifikátor přístupu `protected`. Jaké třídy budou moci přistupovat k operacím a proměnným?

:c1 jakékoliv třídy

:c2 Přístup pouze z dané třídy

:c3 Mohou přistupovat třídy stejného balíku, případně z potomka třídy kdekoliv

:c4 Z kterékoliv třídy stejného balíku

:c3 ok ex

--

K čemu slouží konstruktory

:c1 zadání hodnot instanci třídy

:c2 Konstruuje třídu

:c3 spouští instanci třídy

:c4 Z jedné instance implicitně vytváří jinou instanci třídy

:c1 ok ex

--

Kolik metod (nezděděných) bude mít k dispozici objekt `zaměstnanec`?

```
class Zaměstnanec {  
    public Zaměstnanec(int vek, int plat) {  
        this.vek = vek;  
        this.plat = plat;  
    }  
}
```

```
protected int vek = 1;
public int vratVek() { return vek; }
public void nastavVek(int vek) { this.vek = vek; }

protected int plat = 1;
    public int vratPlat() { return plat; }
public void nastavPlat(int plat) { this.plat = plat; }
public void predstavSe(){
    System.out.println("Muj vek a plat jsou " + vek+"let "+plat+"Korun");
}

public static void main(String[] args) {
    Zamestnanec zamestnanec = new Zamestnanec(30,100);
}
}
```

:r1 1

:r2 2

:r3 3

:r4 4

:r5 5

:r5 ok

--

What kind of comments are in the following codes?

```
/**
```

```
    * Main program class.
```

```
*/
```

```
public class Main {  
  
    public static void main( String[] args ) {  
  
        /* Prints the answer */  
  
        System.out.println( 42 ); // answer is 42  
  
    }  
  
}
```

```
}  
:c1 one line
```

```
:c2 Multiline
```

```
:c3 Documentation
```

```
:c4 Compilation
```

```
:c1:c2:c3 ok ex
```

```
--
```

256 457 can be stored in data types

```
:c1 byte
```

```
:c2 short
```

```
:c3 int
```

```
:c4 long
```

```
:c3:c4 ok ex
```

```
--
```

The Java Virtual Machine task is :

```
:r1 Provide programs always the same environment
```

```
:r2 Compile code
```

```
:r3 GUI for creating a program
```

```
:r1 ok
```

--

The number 315 can be stored in data types

:c1 byte

:c2 short

:c3 int

:c4 long

:c2:c3:c4 ok ex

--

Číslo 2,6 můžeme uložit do datových typů

:c1 int

:c2 long

:c3 float

:c4 double

:c3:c4 ok ex

--

Číslo 256 457 můžeme uložit do datových typů

:c1 byte

:c2 short

:c3 int

:c4 long

:c3:c4 ok ex

--

```
String s3 = s1 + s2;
```

:c1 s1 + s2

:c2 jdk7.0

:c3 jdk

:c4 7.0

:c2 ok ex

--

```
boolean c = a<=b;
```

:c1 true

:c2 false

:c3 chyba

:c1 ok ex

--

```
int y = x++;
```

:r1 0

:r2 1

:r3 2

:r4 3

:r2 ok

--

Hodnota x bude

```
int x=24;
```

```
x /= 2;
```

:r1 24

:r2 2

:r3 12

:r4 chyba

:r3 ok

--

Tento cyklus proběhne:

```
int x = 5;
```

```
while( x >= 0 ) {
```

```
    System.out.println( x );
```

```
    x --;
```

```
}
```

:r1 ani jednou

:r2 2x

:r3 3x

:r4 4x

:r5 5x

:r6 6x

:r6 ok

--

Kolikrát proběhne tento cyklus:

```
int s = 99;
```

```
while( s > 0 ) {
```

```
    if( s%10 == 0 ) {
```

```
        break;
```

```
    }
```

```
    s = s-1;
```

```
    System.out.println( s );
```

```
}
```

:n

:n="9" ok

--

K čemu slouží dědičnost

:c1 koncové (final) metody rozšiřovaná třída zdědí, ale nemůže je překrýt.

:c2 Rodičovská třída dědí všechny metody mateřské třídy

:c3 Rozšiřovaná třída zdědí všechny nesoukromé metody a třídní proměnné předka

:c4 K utváření hierarchie tříd

:c1:c3:c4 ok ex

--

K čemu slouží metody

:r1 K matematickým výpočtům

:r2 Ke statistickým výpočtům

:r3 K několikerému používání stejné posloupnosti příkazů

:r4 K tvorbě objektů

:r3 ok

--

K čemu může sloužit return u metod

:c1 Okamžité ukončení metody

:c2 Vráťí na začátek kódu

:c3 Vráťí hodnotu

:c4 Vráťí na začátek metody

:c1:c3 ok ex