

Laboratorio di Programmazione

27 – 31 Ottobre

Simone Zaccaria

Esercizio di Ripasso

Scrivere un programma che:

1. dichiarare 2 variabili a e b;
2. inizializzare le 2 variabili con valori numerici a scelta;
3. stampare a video il contenuto delle 2 variabili;
4. scambiare il contenuto delle 2 variabili;
5. stampare di nuovo a video il contenuto delle 2 variabili per verificare che i valori siano stati scambiati.

Soluzione Ripasso

```
int a = 1, b = 2;
```

```
System.out.println("a:" + a + "    b:" + b);
```

```
int temp = a;
```

```
a = b;
```

```
b = temp;
```

```
System.out.println("a:" + a + "    b:" + b);
```

Esercizio 1.1

- Un programma che chiede di digitare due numeri interi.
- Legge due numeri interi con la funzione `nextInt()`
- Stampa a video i due numeri letti
- Stampa la loro somma
- Stampa la loro divisione (Attenzione ai valori 0!)

Soluzione 1.1

```
import java.util.Scanner;
public class ScannerDemo {

    public static void main(String[] args) {
        Scanner tastiera = new Scanner(System.in);
        System.out.println("Digita due numeri interi");
        System.out.println("separati da uno o piu' spazi:");
        int n1, n2;
        n1 = tastiera.nextInt();
        n2 = tastiera.nextInt();
        System.out.println("Hai digitato " + n1 + " e " + n2);
        System.out.println("Somma dei numeri " + (n1 + n2));
        System.out.println("«La divisione dei numeri " +
                            (n1 / n2));
    }
}
```

Esercizio 1.2

- Continuare il programma, in modo da:
 - Leggere due numeri double
 - Stamparli a video
 - Stampare a video la loro divisione

Soluzione 1.2

```
System.out.println("Ora digita altri due numeri.");  
System.out.println("E' ammesso anche il separatore  
                                decimale.");
```

```
double d1, d2;
```

```
d1 = tastiera.nextDouble();  
d2 = tastiera.nextDouble();
```

```
System.out.println("Hai digitato " + d1 + " e " + d2);  
System.out.println("«Il loro quoziente e' " + (d1 / d2) );
```

Esercizio 1.3

- Continuare il programma, in modo da:
 - Leggere due parole (separate da spazi)
 - Stamparle a video

Soluzione 1.3

```
System.out.println("Ora digita due parole:");
```

```
String s1, s2;
```

```
s1 = tastiera.next();
```

```
s2 = tastiera.next();
```

```
System.out.println("Hai digitato \"" + s1 + "\"  
e \"" + s2 + "\"");
```

Esercizio 1.4

- Continuare il programma, in modo da:
 - Leggere una frase
 - Stamparla a video

Soluzione 1.4

```
System.out.println("Digita ora una riga di  
                    testo:");
```

```
s1 = tastiera.nextLine();
```

```
System.out.println("Hai digitato: \"" + s1 + "\"");
```

ATTENZIONE!!!!

ATTENZIONE

- Scrivere un programma che:
 - Legge due interi e li stampa
 - Legge una stringa e la stampa

ATTENZIONE

```
int n1, n2;
```

```
System.out.println("Digita ora una riga di testo:");  
n1 = tastiera.nextInt();  
n2 = tastiera.nextInt();
```

```
System.out.println(n1 + " e " + n2);
```

```
System.out.println("Digita ora una stringa:");  
String s = tastiera.nextLine();
```

```
System.out.println("Stringa:" + s);
```

ATTENZIONE

```
int n1, n2;  
  
System.out.println("Digita ora una riga di testo:");  
n1 = tastiera.nextInt();  
n2 = tastiera.nextInt();  
  
System.out.println(n1 + " e " + n2);  
  
System.out.println("Digita ora una stringa");  
String s = tastiera.nextLine();  
  
System.out.println("Stringa:" + s);
```

ATTENZIONE

```
int n1, n2;
```

```
System.out.println("Digita ora una riga di testo:");  
n1 = tastiera.nextInt();  
n2 = tastiera.nextInt();
```

```
tastiera.nextLine(); //Necessario per gestire il \n
```

```
System.out.println(n1 + " e " + n2);
```

```
System.out.println("Digita ora una stringa:");  
String s = tastiera.nextLine();
```

```
System.out.println("Stringa:" + s);
```

Variabili Booleane

```
boolean t = true;
```

```
boolean t = false;
```

```
boolean t = x > 4
```

```
boolean t = 4%2 == 0
```

```
Boolean t = (x%2 == 0) && (x <= 10)
```

Espressioni Booleane

Gli operatori relazionali servono a confrontare due operandi (o due espressioni) e restituiscono un valore booleano (tipo `boolean`).

Operatore	Significato	Esempio
>	maggiore	<code>5>4</code> → <code>true</code>
<	minore	<code>5<4</code> → <code>false</code>
>=	maggiore o uguale	<code>5>=4</code> → <code>true</code>
<=	minore o uguale	<code>5<=4</code> → <code>false</code>
==	uguale	<code>5==4</code> → <code>false</code>
!=	diverso	<code>5!=4</code> → <code>true</code>

Espressioni Booleane

Gli operatori booleani servono a comporre più espressioni booleane e ritornano come risposta un valore booleano.

Operatore	Significato	Esempio
<code>&</code>	and (congiunzione)	<code>true & false → false</code>
<code>&&</code>	and (congiunzione) - short circuit	<code>true && false → false</code>
<code> </code>	or (disgiunzione)	<code>true false → true</code>
<code> </code>	or (disgiunzione) - short circuit	<code>true false → true</code>
<code>!</code>	not (negazione)	<code>!true → false</code>

Short Circuit

Perché avere due diversi operatori per la disgiunzione e la congiunzione?

- Quando valutiamo una disgiunzione e troviamo uno degli operatori vero allora non serve valutare gli altri per sapere che il risultato sarà `true`.
- Quando valutiamo una congiunzione e troviamo uno degli operatori false allora non serve valutare gli altri per sapere che il risultato sarà `false`.

Quindi `op1 & op2` valuta sia `op1` che `op2`.

`op1 && op2` invece, valuta `op1`, se esso è `false` ritorna subito `false` senza valutare `op2`.

Gli operatori `|` e `||` si comportano in modo simile.

Esercizio 2.1

```
int x = 37;  
int y = 42;  
boolean b = x>7 && y < 50;  
System.out.println(b);
```

```
int x = 37;  
int y = 0;  
boolean b = y!=0 && x/y == 2;  
System.out.println(b);
```

Soluzione 2.1

```
int x = 37;  
int y = 42;  
boolean b = x > 7 && y < 50;  
System.out.println(b);
```

Stampa a schermo true.

```
int x = 37;  
int y = 0;  
boolean b = y != 0 && x / y == 2;  
System.out.println(b);
```

Stampa a schermo false visto che $y \neq 0$ è falso **non** valuta $x/y == 2$.
E se avessimo utilizzato $\&$ invece di $\&\&$?

Esercizio 2.2

```
int x = 37;  
int y = 0;  
boolean b = y!=0 & x/y == 2;  
System.out.println(b);
```

Soluzione 2.2

```
int x = 37;  
int y = 0;  
boolean b = y!=0 & x/y == 2;  
System.out.println(b);
```

Si sarebbe ottenuto un errore simile al seguente:

```
Exception in thread "main" java.lang.ArithmeticException: / by zero  
at DivByZero.main(DivByZero.java:6)
```

Questo in quanto anche se $y \neq 0$ restituisce false la seconda parte dell'espressione viene valutata portando ad una divisione per zero.

Precedenze degli Operatori

Gli operatori seguono le seguenti regole di precedenza (da quelli con precedenza maggiore a quelli con precedenza minore):

1 Operatori aritmetici

- 1 - (unario), ++, --
- 2 *, /, %
- 3 +, -

2 Operatori relazionali

- 1 >, <, >=, <=
- 2 ==, !=

3 Operatori booleani

- 1 !
- 2 &
- 3 |
- 4 &&
- 5 ||

Esercizio 3.1

```
5 > 0 && 6 < 7;
```

```
5 != 5 && 7 > 8 | 9 < 10
```

Soluzione 3.1

`5 > 0 && 6 < 7;`

Viene valutato `5 > 0` → `true`.

Viene valutato `6 < 7` → `true`.

Viene valutata la congiunzione e quindi ritornato `true`.

Sarebbe cambiato qualcosa utilizzando `&` invece di `&&`?

`5 != 5 && 7 > 8 | 9 < 10` Viene valutato `7 > 8` → `false` (ricordate che `|` ha precedenza maggiore di `&&`).

Viene valutato `9 < 10` → `true`.

Viene valutata la disgiunzione, quindi `7 > 8 | 9 < 10` → `true`.

Viene valutato `5 != 5` → `false`.

Viene valutata la congiunzione, ritornando, infine, `false`.

Sarebbe cambiato qualcosa utilizzando `||` invece di `|`?

Soluzione 3.1

`5 > 0 && 6 < 7;`

Viene valutato `5 > 0` → `true`.

Viene valutato `6 < 7` → `true`.

Viene valutata la congiunzione e quindi ritornato `true`.

Sarebbe cambiato qualcosa utilizzando `&` invece di `&&`? NO!

`5 != 5 && 7 > 8 | 9 < 10` Viene valutato `7 > 8` → `false` (ricordate che `|` ha precedenza maggiore di `&&`).

Viene valutato `9 < 10` → `true`.

Viene valutata la disgiunzione, quindi `7 > 8 | 9 < 10` → `true`.

Viene valutato `5 != 5` → `false`.

Viene valutata la congiunzione, ritornando, infine, `false`.

Sarebbe cambiato qualcosa utilizzando `||` invece di `|`? SI! (per la precedenza degli operatori il risultato finale sarebbe stato `true`).

Esercizio 3.2

Come esprimere la condizione: *la variabile x contiene un valore pari?* (più risposte possibili)

- ① $x == 2*y$.
- ② $x = 2*y$.
- ③ $2*(x/2) == x$.
- ④ $2*(x/2) = x$.
- ⑤ $x\%2 == 0$.
- ⑥ $x\%2 = 0$.
- ⑦ Nessuna delle precedenti.

Soluzione 3.2

Come esprimere la condizione: *la variabile x contiene un valore pari?* (più risposte possibili)

- 1 $x == 2*y$.
- 2 $x = 2*y$.
- 3 $2*(x/2) == x$.
- 4 $2*(x/2) = x$.
- 5 $x\%2 == 0$.
- 6 $x\%2 = 0$.
- 7 Nessuna delle precedenti.

Le risposte giuste sono $x\%2 == 0$ e $2*(x/2) == x$. Il perché della prima è evidente, la seconda perché vi ricordo che stiamo utilizzando la divisione intera.

Esercizio 3.3

Come esprimere la condizione: *la variabile x contiene un valore positivo multiplo di 3?* (più risposte possibili)

- ① $x > 0 \ \& \ x\%3 == 0.$
- ② $x > 0 \ \&\& \ x\%3 == 0.$
- ③ $x > 0 \ \&\& \ x\%3 != 0.$
- ④ $x > 0 \ | \ x\%3 == 0.$
- ⑤ $x > 0 \ || \ x\%3 == 0.$
- ⑥ $x > 0 \ || \ x\%3 != 0.$
- ⑦ Nessuna delle precedenti.

Soluzione 3.3

Come esprimere la condizione: *la variabile x contiene un valore positivo multiplo di 3?* (più risposte possibili)

- ① `x > 0 & x%3 == 0.`
- ② `x > 0 && x%3 == 0.`
- ③ `x > 0 && x%3 != 0.`
- ④ `x > 0 | x%3 == 0.`
- ⑤ `x > 0 || x%3 == 0.`
- ⑥ `x > 0 || x%3 != 0.`
- ⑦ Nessuna delle precedenti.

Le risposte giuste sono

`x > 0 && x%3 == 0`

e

`x > 0 & x%3 == 0.`

Esercizio 3.4

Come esprimere la condizione: *la variabile x contiene un valore positivo multiplo di 3 ma non di sette?* (più risposte possibili)

- ① $x > 0 \ \& \ x\%3 == 0 \ || \ !(x\%7 == 0).$
- ② $x > 0 \ \& \ x\%3 == 0 \ \&\& \ !(x\%7 == 0).$
- ③ $x > 0 \ \&\& \ x\%3 == 0 \ || \ x\%7 != 0.$
- ④ $x > 0 \ \&\& \ x\%3 == 0 \ \&\& \ x\%7 != 0.$
- ⑤ $x > 0 \ || \ (x\%3 == 0 \ \&\& \ x\%7 == 0).$
- ⑥ $x > 0 \ \&\& \ x\%21 != 0.$
- ⑦ Nessuna delle precedenti.

Soluzione 3.4

Come esprimere la condizione: *la variabile x contiene un valore positivo multiplo di 3 ma non di sette?* (più risposte possibili)

- ① $x > 0 \ \& \ x\%3 == 0 \ || \ !(x\%7 == 0)$.
- ② $x > 0 \ \& \ x\%3 == 0 \ \&\& \ !(x\%7 == 0)$.
- ③ $x > 0 \ \&\& \ x\%3 == 0 \ || \ x\%7 != 0$.
- ④ $x > 0 \ \&\& \ x\%3 == 0 \ \&\& \ x\%7 != 0$.
- ⑤ $x > 0 \ || \ (x\%3 == 0 \ \&\& \ x\%7 == 0)$.
- ⑥ $x > 0 \ \&\& \ x\%21 != 0$.
- ⑦ Nessuna delle precedenti.

Le risposte giuste sono

$x > 0 \ \&\& \ x\%3 == 0 \ \&\& \ x\%7 != 0$ e $x > 0 \ \& \ x\%3 == 0 \ \&\& \ !(x\%7 == 0)$.

Esercizio 3.5

Come esprimere la condizione: *la variabile x è un multiplo intero del valore della variabile y oppure è multiplo di 11 e 19?* (più risposte possibili)

- 1 $x\%y == 0 \ \&\& \ x\%11 == 0 \ \&\& \ x\%19 == 0.$
- 2 $x\%y == 0 \ || \ x\%(11*19) == 0.$
- 3 $x\%y == 0 \ \&\& \ (x\%11 == 0 \ || \ x\%19 == 0).$
- 4 $x\%y == 0 \ || \ (x\%11 == 0 \ \&\& \ x\%19 == 0).$
- 5 $x\%y == 0 \ \&\& \ x\%(11*19) == 0.$
- 6 $y\%x == 0 \ \&\& \ x\%11 == 0 \ \&\& \ x\%19 == 0.$
- 7 Nessuna delle precedenti.

Esercizio 3.5

Come esprimere la condizione: *la variabile x è un multiplo intero del valore della variabile y oppure è multiplo di 11 e 19?* (più risposte possibili)

- ① $x\%y == 0 \ \&\& \ x\%11 == 0 \ \&\& \ x\%19 == 0$.
- ② $x\%y == 0 \ || \ x\%(11*19) == 0$.
- ③ $x\%y == 0 \ \&\& \ (x\%11 == 0 \ || \ x\%19 == 0)$.
- ④ $x\%y == 0 \ || \ (x\%11 == 0 \ \&\& \ x\%19 == 0)$.
- ⑤ $x\%y == 0 \ \&\& \ x\%(11*19) == 0$.
- ⑥ $y\%x == 0 \ \&\& \ x\%11 == 0 \ \&\& \ x\%19 == 0$.
- ⑦ Nessuna delle precedenti.

Le risposte giuste sono

$x\%y == 0 \ || \ (x\%11 == 0 \ \&\& \ x\%19 == 0)$

e

$x\%y == 0 \ || \ x\%(11*19) == 0$.

Esercizio 4

Scrivere un programma Java che risolva le seguenti espressioni e ne stampi il risultato:

1. `4>6 && 5<9 || 7%6==1`
2. `2<=3 || 10==10 && 7!=4 || 2>-1`
3. `(a & b) ∨ (¬ a & b)`
4. `¬(¬ a & b) ∨ c`
5. `c == 'x' && b == true`
6. `c == 'x' && b`
7. `!b || a<length`
8. `b<=5 && a > 5` assegnando $a = 6$ e $b = 5$
9. `b<=5 || a > 5` assegnando $a = 6$ e $b = 5$
10. `a == b++` assegnando $a = 5$ e $b = 5$
11. `a==++b` assegnando $a = 5$ e $b = 5$

Dove necessario, definire delle opportune variabili con valori di inizializzazione a piacere. Scegliere un tipo di dato adeguato per il risultato.

if

La selezione serve a scegliere se eseguire o no una o più operazioni in seguito ad una condizione:

```
if (x%2 == 0)
    System.out.println("x e' pari.");
```

Stampa a schermo x e' pari solo se $x\%2==0$ è true.
Più in generale la forma del costrutto if è:

```
if ( /* espressione booleana */ )
    /* Istruzione da eseguire se l'espressione e' true */
```

If...else

Se vogliamo scegliere tra due istruzioni a seconda di una condizione possiamo utilizzare i costrutti `if ... else`:

```
if (x%2 == 0)
    System.out.println("x e' pari.");
else
    System.out.println("x e' dispari");
```

Stampa a schermo `x e' pari` solo se `x%2==0` è true *altrimenti* stampa `x e' dispari`.

Più in generale la forma del costrutto `if ... else` è:

```
if (/*espressione booleana*/)
    /*Istruzione da eseguire se l'espressione e' true*/
else
    /*Istruzione da eseguire se l'espressione e' false*/
```

Esercizio 5

- Scrivere un programma che:
 - Inizializza una variabile con un valore casuale
 - Testa se la variabile è minore di 7:
 - Se è maggiore di sette allora stampa la frase «x è minore di 7»
 - Altrimenti stampa la frase «x è maggiore o uguale a 7»

Soluzione 5

Il seguente codice:

```
int a = 5;
if (a < 7)
    System.out.println("a e' minore di 7");
else
    System.out.println("a e' maggiore o uguale a 7");
System.out.println("Questa frase viene stampata");
```

stampa:

a e' minore di 7

Questa frase viene stampata

Parentesi!!

Il costrutto `if` visto fino ad ora permette di eseguire una sola istruzione... però non è pratico!

Possiamo quindi “raggruppare” più istruzioni tra parentesi graffe:

```
if (x != 5) {  
    System.out.println("x vale " + x);  
    x = x+3;  
} else {  
    System.out.println("x vale 5");  
    x = x-2;  
}
```

Annidamenti

Possiamo utilizzare `if` dentro ad un `if`:

```
if (x < 10)
    if (x > 5)
        System.out.println("x e' maggiore di 5 e minore↵
                               di 10");
```

Notate come il costrutto `if` nella sua interezza conti come una singola istruzione.

Annidamenti

```
If (x > 10) {  
    if (x > 5) {  
        System.out.println(«Tra 6 e 10»);  
    } else {  
        System.out.println(«Tra 0 e 5»);  
    }  
} else {  
    if (x > 15) {  
        ...  
    } else {  
        ...  
    }  
}
```

Switch cases

```
If (x > 100) {  
    ...  
} else if (x > 50) {  
    ...  
} else if (x > 20) {  
    ...  
} else if (x > 10) {  
    ...  
} else if (x >= 0) {  
    ...  
} else {  
    ...  
}
```

Annidamenti - Ambiguità

Se avete più if seguiti da un else bisogna sapere a quale if è associato l'else:

```
if (x < 10)
    if (x > 5)
        System.out.println("x e' maggiore di 5 e minore
                             di 10");
    else
        System.out.println("Quando viene eseguito l'
                             else?");
```

L'else è associato al primo if privo di else che lo precede, quindi a `if (x > 5)`

Annidamenti - Ambiguità

Possiamo sempre utilizzare le parentesi per cambiare il comportamento:

```
if (x < 10) {  
    if (x > 5)  
        System.out.println("x e' maggiore di 5 e minore ←  
        di 10");  
} else  
    System.out.println("Quando viene eseguito l'else?") ←  
    ;
```

In questo modo l'else è associato a `if (x < 10)`.

Esercizio 6.1

Cosa stampa il seguente codice?

```
int x = 10;
int y = 5;
if (x%y == 0)
    System.out.println("ramo if");
else
    System.out.println("ramo else");
```

Soluzione 6.1

Cosa stampa il seguente codice?

```
int x = 10;
int y = 5;
if (x%y == 0)
    System.out.println("ramo if");
else
    System.out.println("ramo else");
```

Stampa ramo if in quanto $x\%5 == 0$ è true.

Esercizio 6.2

Cosa stampa il seguente codice?

```
int x = 7;
int y = 6;
if (x > y) {
    x--;
    if (x > y)
        System.out.print(" ciao");
    else
        System.out.print(" addio");
}
System.out.println(" mondo");
```

Soluzione 6.2

Cosa stampa il seguente codice?

```
int x = 7;
int y = 6;
if (x > y) {
    x--;
    if (x > y)
        System.out.print(" ciao");
    else
        System.out.print(" addio");
}
System.out.println(" mondo");
```

Stampa addio mondo in quanto $x > y$ è true la prima volta ma false la seconda in quanto x viene decrementato dopo il primo if.

Esercizio 6.3

Dove è l'errore nel seguente codice?

```
int x = 7;  
if (x > 6)  
    x--;  
    System.out.println("ramo if");  
else  
    System.out.println("ramo else");
```

Soluzione 6.3

Dove è l'errore nel seguente codice?

```
int x = 7;  
if (x > 6)  
    x--;  
    System.out.println("ramo if");  
else  
    System.out.println("ramo else");
```

Vi sono **due** istruzioni dopo l'if e non sono racchiuse tra parentesi graffe. Quindi l'else successivo non è associato ad alcun if.

Esercizio 6.4

Cosa stampa il seguente codice?

```
int x = 7;  
if (x > 6)  
if (x < 5)  
System.out.print("stampa 1");  
else  
System.out.print("stampa 2");
```

Soluzione 6.4

Cosa stampa il seguente codice?

```
int x = 7;  
if (x > 6)  
if (x < 5)  
System.out.print("stampa 1");  
else  
System.out.print("stampa 2");
```

Stampa stampa 2 Infatti $x > 6$ è true quindi si entra nel blocco `if ... else` successivo (ricordate che l'`else` viene associato al primo blocco `if` precedente privo di `else`).

Esercizio 6.5

```
if (x > 0)
    x = 3;
else
    if (x < 0)
        x = x + 2;
    else
        x = 5;
```

```
if (x > 0)
    x = 3;
if (x < 0)
    x = x + 2;
if (x == 0)
    x = 5;
```

- Per quali casi i programmi non hanno lo stesso comportamento?

Esercizio 6.5

```
if (x>0)
    x = 3;
else
    if (x<0)
        x = x+2;
    else
        x = 5;
```

```
if (x>0)
    x = 3;
if (x<0)
    x = x+2;
if (x==0)
    x = 5;
```

- Per quali casi i programmi non hanno lo stesso comportamento?
RISPOSTA: per $x = -2$

Esercizio 6.5

```
if (x > 0)
    x = 3;
else
    if (x < 0)
        x = x + 2;
    else
        x = 5;
```

```
if (x > 0)
    x = 3;
if (x < 0)
    x = x + 2;
if (x == 0)
    x = 5;
```

- Per quali casi i programmi non hanno lo stesso comportamento?
RISPOSTA: per $x = -2$
- È possibile che nel secondo programma vengano eseguite tutte e tre le istruzioni nei blocchi if?

Esercizio 6.5

```
if (x > 0)
    x = 3;
else
    if (x < 0)
        x = x + 2;
    else
        x = 5;
```

```
if (x > 0)
    x = 3;
if (x < 0)
    x = x + 2;
if (x == 0)
    x = 5;
```

- Per quali casi i programmi non hanno lo stesso comportamento?
RISPOSTA: per $x = -2$
- È possibile che nel secondo programma vengano eseguite tutte e tre le istruzioni nei blocchi if?
RISPOSTA: no.

Esercizio 6.6

1. Verificare che i due programmi seguenti non sono equivalenti

Programma a

```
if(x>0)
    x = 3;
else
    if(x<0)
        x = x+2;
    else
        x = 5;
```

Programma b

```
if(x>0)
    x = 3;
if(x<0)
    x = x+2;
if (x==0)
    x = 5;
```

- In quali condizioni i programmi non hanno lo stesso comportamento?
- È possibile che nel secondo programma vengano eseguite tutte e tre le istruzioni nei blocchi if?

Esercizio Completi 1

- Scrivere un programma che chiede di inserire due valori e ne calcola il minimo, stampandolo a schermo.

Esercizi Completi 2

- Scrivere un programma che chiede di inserire tre valori e ne calcola il minimo, stampandolo a schermo.

Esercizi Completi 3

Scrivere un programma che inizializzata una variabile intera, se il suo valore

- è dispari lo stampi
- è pari stampi la sua metà
- Fare l'esercizio inserendo tutte le istruzioni necessarie nel blocco if ({}).
- Opzionale. Provare a farlo senza utilizzare le parentesi graffe (inserendo un'unica istruzione nell'if)

Esercizi Completi 4

Scrivere un programma che inizializzate due variabili numeriche

- se la seconda è $\neq 0$, effettui la loro divisione intera
- altrimenti stampi un messaggio di errore
- Cosa succede se si effettua la divisione con un divisore $= 0$?

Esercizi Completi 5

6. Scrivere un programma che, inizializzata una variabile con un numero compreso tra 0 e 100,
- stampi il messaggio “Very Low” se $0 \leq x < 20$
 - stampi il messaggio “Low” se $20 \leq x < 40$
 - stampi il messaggio “Normal” se $40 \leq x < 60$
 - stampi il messaggio “High” se $60 \leq x < 80$
 - stampi il messaggio “Very High” altrimenti

Esercizi Completi 6

Scrivete un programma Java che:

- Chieda all'utente nome e cognome (con opportuni messaggi utente);
- Stampi a video nome e cognome (separati da uno spazio) con le iniziali in maiuscolo (anche se l'utente le ha scritte in minuscolo).

Aiuti: la chiamata per convertire le stringhe in maiuscolo è `toUpperCase()` e prende come parametro una stringa.

Esercizi Completi 7

Si scriva un programma in Java che, utilizzando gli appropriati metodi sulle stringhe chieda all'utente una frase e:

- se la frase contiene 2 o più parole:
 - stampi a video che la frase contiene due parole o più
 - stampi a video la prima parola e la sua lunghezza
 - stampi a video l'ultima parola della frase e la sua lunghezza
- se la frase contiene una parola:
 - stampi a video che la frase contiene una sola parola
 - stampi a video la parola e la sua lunghezza
- se la frase inserita e' vuota:
 - stampi a video che la frase non contiene parole.

Si tenga conto, e si gestisca in modo appropriato, l'inserimento da parte dell'utente di spazi o tab all'inizio e alla fine della frase.

AIUTO/Semplificazione: La frase non contiene caratteri di punteggiatura o tab; perciò le parole sono individuabili semplicemente 'ricercando' lo spazio.