

Java secondo contatto

Nel tunnel una luce...






Esercizio - Contatore

- Definire la classe Contatore con le seguenti caratteristiche:
 - Il metodo getValore ritorna il valore attuale (intero)
 - Il metodo contaSu incrementa il valore del contatore e non ritorna nulla
- Definire la classe Avvia (il main) per
 - Creare una istanza Contatore;
 - Incrementare il contatore 2 volte;
 - Stamparne il valore.



Classe Contatore

```
public class Contatore {  
    int valore = 0; // var. istanza  
    public int getValore() { // no parametri  
        return valore;  
    }  
    public void contaSu() {  
        // non restituisce info  
        valore = valore + 1; // valore++;  
    }  
}
```



Metodo main – classe separata

```
public class Avvia {  
    public static void main(String[] args) {  
        Contatore c = new Contatore();  
        c.contaSu();  
        c.contaSu();  
        System.out.println(c.getValore());  
    }  
}
```

→ Si poteva scrivere il main direttamente nella classe Contatore?

Struttura di una classe

```
public class Contatore {
```

Definizione classe

```
    int valore = 0;
```

Definizione var. di classe

Inizializzazione var. di classe

```
    public int getValore() {
```

Definizione metodo

```
        return valore;
```

Tipo di ritorno del metodo

```
    }
```

Uso var. di istanza della classe

```
    public void contaSu() {
```

```
        valore = valore + 1;
```

```
    }
```

```
}
```

Indentazione e campo di azione

```
public class Contatore
```

```
{
```

Campo

```
int valore = 0;
```

Metodo

```
public int getValore()
```

```
{
```

```
.....
```

```
}
```

Metodo

```
public void contaSu()
```

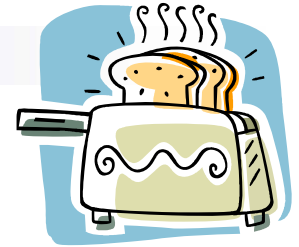
```
{
```

```
.....
```

```
}
```

```
}
```

Dichiarazione di classe



Aggiungiamo altri metodi

- Aggiungere due metodi:
 - void **contaGiu()** che decrementa il contatore di 1 unità
 - void **impostaValore(int val)** che imposta il valore del contatore al valore val
- Definiamo una nuova classe Avvia:
 - Creare una istanza Contatore;
 - stamparne il valore
 - Incrementarlo e stamparne il valore
 - Decrementarlo e stamparne il valore
 - Impostare il valore a 10 e stampare il valore

Classe Contatore - nuovi metodi

```
public class Contatore {  
    private int valore = 0; // var. istanza  
    public int getValore() { // no parametri  
        return valore;  
    }  
    public void contaSu() { // non resti tue sce info  
        valore = valore + 1; // valore++;  
    }  
  
    public void contaGiù() {  
        valore = valore - 1; //valore --  
    }  
    public void impostaValore(int val) {  
        valore=val ;  
    }  
}
```

Parametro del metodo



Metodo main aggiornato

```
public class Avvia {  
    public static void main(String[] args) {  
        Contatore c = new Contatore();  
        System.out.println(c.getValore());  
  
        c.contaSu();  
        System.out.println(c.getValore());  
  
        c.contaGiù();  
        System.out.println(c.getValore());  
  
        c.impostaValore(10);  
        System.out.println(c.getValore());  
    }  
}
```



Inizializzazione del contatore

- Come creare l'oggetto contatore con valore iniziale pari a 10?
 - Modifico l'assegnazione del campo della classe

```
private int valore = 10;
```

 - E se volessi usare due contatori, con valori 10 e 20?
 - Utilizzo il metodo impostaValore(...)

```
Contatore c = new Contatore();  
C. impostaValore(10); ...
```

Classe Contatore e costruttori

```
public class Contatore {  
    int valore;  
    . . . . .
```

//costruttore

```
public Contatore(int v){  
    valore = v;  
}
```

Parametro del metodo

Var. di istanza /
campo della classe

//costruttore senza argomenti

```
public Contatore(){  
    this.valore = 0;  
}  
}
```

Riferimento Implicito this



Metodo main

```
public class Avvia {  
    public static void main(String[] args) {  
        Contatore c = new Contatore(10);  
        System.out.println(c.getValore());  
        c.contaGiù();  
        System.out.println(c.getValore());  
    }  
}
```

Utilizzo di piu' istanze



- Creare due istanze di Contatore con valori iniziali pari a 10 e 20 rispettivamente
- Incrementare il contatore di ognuno e stamparne il valore



Metodo main - Uso di più istanze

```
public class Avvia {  
    public static void main(String[] args) {  
        Contatore c1 = new Contatore(10);  
        Contatore c2 = new Contatore(20);  
        c1.contaSu();  
        c2.contaSu();  
        System.out.println("Primo: " + c1.getValore());  
        System.out.println("Secondo: " + c2.getValore());  
    }  
}
```



Contatore che ricorda

- Modificare la classe contatore in modo che
 - Venga memorizzato il valore iniziale impostato
 - Contenga un metodo **reset()** che permette di reimpostare il valore del contatore al valore iniziale
 - Contenga un metodo **getValoreIniziale()** che ritorna il valore iniziale

Esercizio per Casa

- Classe Vettore (vettore geometrico con origine nel centro degli assi)

```
public class Vettore{
    //crea un vettore con ascissa e ordinata specificati
    public Vettore(double x, double y){...}
    //ritorna il valore dell'ascissa
    public double getAscissa(){...}
    //ritorna il valore dell' ordinata
    public double getOrdinata(){...}
    //stampa su out il vettore nella forma (x,y)
    public void stampa() {...}
    //somma al vettore corrente quello passato come parametro e modifica l'oggetto
    public void somma(Vettore f) {...}
    //calcola il modulo del vettore
    public double calcolaModulo() {...}
    //moltiplica il vettore corrente per uno scalare
    public double moltiplicaPerScalare(double s) {...}
}
```


Competenze

- Utilizzo delle classi
 - Creazione di oggetti
 - Definizione di metodi
 - Invocazione di metodi
 - Definizione di campi
- Utilizzo di classi multiple
- Definizione di classi (fabbrica di oggetti)
 - Metodi di classe
 - Variabili di istanza (campi)
 - Metodi con parametri
- Creazione di istanze (oggetti)
- Costruttori

