

# **Esercizi della lezione 5 di Java**

## Esercizio 5

Create il **tipo di dato Counter** dell'Esercizio 1 come **sottoclasse del tipo di dato SimpleCounter**.

## Esercizio 1

Create un **tipo di dato Counter** che abbia:

**un valore attuale**

**un valore massimo di conteggio**

**uno stato interno che indica se si è verificato un errore.** Il valore massimo è selezionabile dall'utente alla costruzione. Se si tenta di superarlo, viene modificato lo stato del contatore per memorizzare l'avvenuta condizione d'errore e le successive operazioni di modifica non devono avere effetto

Il contatore deve offrire **i seguenti metodi:** `void inc()`, `void reset()`, `int getValue()`, `boolean isError()`

Infine, **scrivete un'applicazione Java** che:

Crea e inizializza un nuovo Counter con valore massimo n  
Incrementa il contatore per n+1 volte e visualizza, ogni volta, il valore attuale e lo stato interno (errato/corretto)

Terminare

## Tipo di dato SimpleCounter

```
public class SimpleCounter  
{ private int valore;  
  
public void inc(){  
  valore=valore+1;}  
  
public int get(){  
  return(valore);}  
  
public void reset(){  
  valore=0;}  
}
```

## Soluzione Esercizio 5

```
public class Counter extends SimpleCounter{
```

```
    /*la variabile valore non la dichiaro: è ereditata dalla superclasse  
    *Counter. Non posso però accedere ad essa poichè è privata  
    *semplicemente chiamandola. La posso però utilizzare invocando i metodi  
    della classe Counter*/
```

```
    private int valore_massimo;  
    private boolean errore=false;
```

```
    //Costruttore
```

```
    public ContEst(int vm){
```

```
    /* Se non scrivo niente viene invocato il costruttore anonimo della  
    superclasse . Se invece voglio invocare un costruttore della superclasse  
    con dei parametri scriverò così: super(val_iniz) */
```

```
    valore_massimo=vm;  
    errore=false; }
```

*//Metodo per incrementare il valore*

```
public void inc(){  
if (errore) return;
```

*/\* per invocare la variabile valore nella superclasse poichè essa è privata, la invoco con il metodo get sull'oggetto corrente: this. \*/*

```
if (this.get()==valore_massimo)  
{errore=true;}
```

*// richiamo il metodo inc definito nella superclasse*

```
else super.inc(); }
```

*//Metodo per resettare il valore*

```
public void reset(){
```

*//richiamo il metodo reset definito nella superclasse*

```
super.reset();
```

```
errore=false;}
```

*//Metodo per ottenere informazioni sullo stato interno*

```
public boolean isError(){  
return errore;}
```

*//Metodo main*

```
public static void main(String[] args) {  
    Counter c= new Counter(5);  
    for(int i=0; i<6; i++){  
        c.inc();  
        System.out.println("Valore del contatore: " + c.get());  
        System.out.println("Stato di errore: " + c.isError());  
    }  
}  
}
```

## E se scrivessimo il main così?

*//Metodo main*

```
public static void main(String[] args) {  
    SimpleCounter c= new Counter(5);  
    for(int i=0; i<6; i++){  
        c.inc();  
        System.out.println("Valore del contatore: " + c.get());  
        System.out.println("Stato di errore: " + c.isError());  
    }  
}  
}
```

**E' giusto?**

**c.inc() quale metodo richiama?**

**Cosa ci aspettiamo in output?**

## E se invece?

*//Metodo main*

```
public static void main(String[] args) {  
    Counter c= new SimpleCounter(5);  
    for(int i=0; i<6; i++){  
        c.inc();  
        System.out.println("Valore del contatore: " + c.get());  
        System.out.println("Stato di errore: " + c.isError());  
    }  
}  
}
```

**E' giusto?**



## Esercizio 6

Create il **tipo di dato Impiegato** come estensione del **tipo di dato Persona**.

Dove una **classe Persona** ha le **variabili nome, cognome, età**. Ha i **metodi GetNome, GetCognome, GetEtà** e un **metodo dettagli()** che restituisce in una stringa le informazioni sulla persona in questione.

Dove una **classe Impiegato** ha le **variabili nome, cognome, età, salario**. Ha un **metodo dettagli()** che restituisce in una stringa le informazioni sulla persona in questione.  
Ha un **metodo aumentasalario()** che aumenti lo stipendio secondo una certa percentuale.

## Tipo di dato Persona

```
public class Persona{  
private String nome;  
private String cognome;  
private int età;
```

```
public Persona(String n,String c, int e){  
nome=n;cognome=c;età=e;}
```

```
public String GetNome()  
{return(nome);}
```

```
public String GetCognome()  
{return(cognome);}
```

```
public int GetEtà()  
{return(età);}
```

```
public String Dettagli()  
{return("Nome= "+nome+" Cognome= "+cognome+" Anni “ + età);} }
```