
Il codice sotto riportato, contiene un errore, qual'è?

```
public interface Animale{ public int getZampe(){} }  
public class Cane implements Animale{ public String getNome(){return "6";} } [1]
```

1. la classe Cane deve implementare il metodo int getZampe()
2. la classe Cane deve implementare il metodo int getNome()
3. l'interfaccia Animale contiene un metodo non implementato
4. l'interfaccia Cane contiene un metodo non implementato

quale è il costrutto per dichiarare un'interfaccia? [1]

1. interface
2. extends
3. implements
4. new

Eseguendo il codice sotto riportato, che valore assumerà l'attributo di istanza "nZampe"?

```
public class Cane{  
    int nZampe;  
    public Cane(int nZampe){this.nZampe=nZampe;}  
    public Cane(){this.nZampe=4;}  
    public static void main(String[] args){  
        Cane billy = new Cane();  
    }  
} [2]
```

1. 4
2. 0
3. null
4. non esiste un costruttore della classe che accetti in input nessun parametro

Eseguendo il codice sotto riportato, quale sarà il contenuto della variabile "indice"?

```
String testo="La divina commedia";  
int indice = testo.indexOf("i"); [1]
```

1. 4
2. 5
3. 3
4. 16

Eseguendo il codice sotto riportato, quale sarà il contenuto della variabile "numero"?

```
String testo="La divina commedia";  
int numero = testo.length(); [1]
```

1. 18
2. 17
3. -1
4. 19

Quale dei seguenti codici è corretto per creare un costruttore della classe sotto riportata?

```
public class Cane{ int nZampe; } [1]
```

1. `public Cane(int nZampe){this.nZampe=nZampe;}`
 2. `public void Cane(int nZampe){this.nZampe=nZampe;}`
 3. `public int Cane(){}`
 4. `new Cane(){}`
-

Eseguendo il codice sotto riportato, quale sarà il contenuto della variabile "indice"?

```
String testo="La divina commedia";  
int indice = testo.indexOf("ia"); [1]
```

1. 16
 2. 4
 3. 5
 4. 3
-

Eseguendo il codice sotto riportato, quale sarà il contenuto della variabile "indice"?

```
String testo="La divina commedia";  
int indice = testo.indexOf("eia"); [1]
```

1. -1
 2. 16
 3. 4
 4. 0
-

Eseguendo il codice sotto riportato, quale sarà il contenuto della variabile "sqrt"?

```
double sqrt = Math.sqrt(Math.min(9,4)); [1]
```

1. 2.0
 2. 3.0
 3. 9.0
 4. 4.0
-

Eseguendo il codice sotto riportato, che valore viene restituito dal metodo `getNome` chiamato su un'istanza della classe `Cane`

```
public interface Animale{ public int getZampe(){ } }  
public class Cane implements Animale{ public String getNome(){return "6"; } } [1]
```

1. viene restituito un errore in fase di compilazione
 2. 6
 3. 12
 4. 3
-

Quale dei seguenti codici è corretto per creare un costruttore della classe sotto riportata?

```
public class Gatto{ } [1]
```

1. `public Gatto(){}`
 2. `public void gatto(){}`
 3. `public int Gatto(){}`
 4. `public gatto(){}`
-

Eseguendo il codice sotto riportato, quale sarà il contenuto della variabile "max"?

```
int max = Math.max(4,9); [1]
```

1. 9
2. 2
3. 4
4. 0

Eseguendo il codice sotto riportato, quale sarà il contenuto della variabile "max"?

```
int max = Math.min(3,6); [1]
```

1. 3
2. 6
3. 18
4. 0

Eseguendo il codice sotto riportato, che valore assumerà l'attributo di istanza "risultato"?

```
public class Somma{
    int risultato;
    public int somma(int a, int b){return a*b;}
    public int somma(int a, int b, int c){return somma(somma(a,b),c);}
    public static void main(String args[]){
        Somma s = new Somma();
        s.risultato = s.somma(2,3,5);
    }
} [2]
```

1. 30
2. 10
3. 25
4. 17

Eseguendo il codice sotto riportato, che valore viene restituito dal metodo getNumero chiamato su un'istanza della classe SottoClasse?

```
public class SuperClasse{ public int getNumero(){return 1;} }
public class SottoClasse extends SuperClasse{ public int getNumero(){return 5;} } [1]
```

1. 5
2. 1
3. 3
4. non è possibile effettuare la chiamata al metodo

Eseguendo il codice sotto riportato, quali attributi di istanza sarà in possesso la classe Studente?

```
public class Persona{int eta;}
public class Studente extends Persona{String nome;} [1]
```

1. int eta; String nome;
2. int eta;
3. String nome;
4. String eta; int nome;

Eseguendo il codice sotto riportato, quale sarà il contenuto dell'attributo di istanza "nomi"?

```
public class Persone{
    ArrayList nomi;
    public Persone(){nomi = new ArrayList();}
    public void aggiungi(String nome){nomi.add(nome);}
    public void rimuovi(int posizione){nomi.remove(posizione);}
}
```

```
public static void main(String args[]){
    Persone p = new Persone();
    p.aggiungi("Luca");
    p.aggiungi("Giovanni");
    p.rimuovi(0);
}
} [2]
```

1. {"Giovanni"}
2. {"Luca", "Giovanni"}
3. {"Luca"}
4. null

Come si definisce una classe che gestisca eccezioni personalizzate? [2]

1. public class EccezioneClasse extends Exception {}
2. public class EccezioneClasse implements Exception {}
3. public class EccezioneClasse exception Exception {}
4. public class EccezioneClasse new Exception {}

un'interfaccia deve avere? [1]

1. tutti i metodi non implementati
2. almeno un costruttore
3. almeno una attributo di istanza
4. almeno una sottoclasse che la estenda

Eseguendo il codice sotto riportato, quale sarà il contenuto dell'attributo di istanza "nomi"?

```
public class Persone{
    ArrayList nomi;
    public Persone(){nomi = new ArrayList();}
    public void aggiungi(String nome){nomi.add(nome);}
    public static void main(String args[]){
        Persone p = new Persone();
        p.aggiungi("Luca");
        p.aggiungi("Giovanni");
    }
} [1]
```

1. {"Luca", "Giovanni"}
2. {"Luca"}
3. {"Giovanni"}
4. null

quale è il costrutto per implementare un'interfaccia? [1]

1. implements
2. extends
3. exception
4. new

Quale è il costrutto per estendere una classe? [1]

1. extends

2. implements
3. exception
4. new

Eseguendo il codice sotto riportato, che valore assumerà l'attributo di istanza "risultato"?

```
public class Divisione{
    int risultato;
    public int divisione(int a, int b){return a/b;}
    public int divisione(int a, int b, int c){return divisione(divisione(a,b),c);}
    public static void main(String args[]){
        Divisione d = new Divisione();
        this.risultato = d.divisione(6,2,3);
    }
} [2]
```

1. 1
2. 3
3. 7
4. 36

Eseguendo il codice sotto riportato, che valore viene restituito dal metodo getNumero chiamato su un'istanza della classe SuperClasse

```
public class SuperClasse{ public int getNumero(){return 33;} }
public class SottoClasse extends SuperClasse{ public int getNumero(){return 6;} } [1]
```

1. 33
2. 6
3. 12
4. non è possibile effettuare la chiamata al metodo

Eseguendo il codice sotto riportato, che valore assumerà l'attributo di istanza "anni"?

```
public class Persona{
    int anni;
    public Persona(int anni){this.anni=anni;}
    public Persona(){this.anni=4;}
    public static void main(String[] args){
        Persona marco = new Persona(12);
    }
} [2]
```

1. 12
2. 4
3. 0
4. non esiste un costruttore della classe che accetti in input un parametro

Eseguendo il codice sotto riportato, che valore assumerà l'attributo di istanza "anni"?

```
public class Persona{
    int anni;
    public Persona(){this.anni=4;}
    public static void main(String[] args){
        Persona marco = new Persona(32);
    }
} [2]
```

1. non esiste un costruttore della classe che accetti in input un parametro
2. 32

3.0

4.4

Eseguendo il codice sotto riportato, quali attributi di istanza sarà in possesso la classe Cane?

```
public class Animale{int nZampe;}  
public class Cane extends Animale{String nome;} [1]
```

1. int nZampe; String nome;
2. int nZampe;
3. String nome;
4. nessun attributo

Qual'è il codice corretto per permettere a un metodo di lanciare un'eccezione MyException in caso di errore? [2]

1. public int calcola() throws MyException
2. public int calcola() implements MyException
3. public int calcola() exception MyException
4. public int calcola() try MyException

Eseguendo il codice sotto riportato, quale sarà il contenuto della variabile "testo"?

```
String testo="sono uno studente che sta facendo il test!";  
testo = testo.replace(' ', '').replace('a', 'o'); [1]
```

1. "sonounostudentechestofocendoiltest!"
2. "sono uno studente che sta facendo il test!"
3. "sonounostudentechestafacendoiltest!"
4. "sono uno studente che sto focendo il test!"