Crocette dicembre 2020

2 Static
Si consideri il seguente codice

```
class A {
 public static int value = 0;
 public static void main (String[] args){
  A.value++;
  A obj = new A():
  }
  }
```

Quale delle seguenti affermazioni è corretta
- Il valore obj.value è 1 alla fine dell'esecuzione del metodo main
- Il valore di A.value è 1 alla fine dell'esecuzione del metodo main

3 this
Si consideri il codice seguente

```
class A {
        in value = 1;
        public A (int value){
         this.value = value;
        }
 }
```

Quale tra le affermazioni seguenti è sbagliata?
- this.value=value assegna un valore al campo value definito nella classe A
- this.value=value assegna un valore al parametro value del costruttore
- La classe A definisce un campo value, e un costruttore che riceve un unico parametro di tipo int

4. princioio di sostituzione
Il principio di sostituzione afferma che un oggetto o1 istanza di una classe C1 può sostituire un oggetto 2 di una classe C2 se:
- La classe C2 definisce un insieme di campi più ampio di quello definito dalla classe C1
- La classe C2 definisce un insieme di campi e metodi più ampio di quello dfinito dalla classe C1
- La classe C1 definisce un insieme di campi e metodi più ampio di quello definito dalla classe C2
- La classe C1 definisce un insieme di campi più ampio di quello quello definito dalla classe C2

5 Stringhe
Si consideri il codice seguente:

```
String s = "abc" + "def";
```

Quale delle seguenti affermazioni è corretta?
- Il programma non compila in quanto non si possono sommare due stringe costanti
- Il programma non compila in quanto le stringhe vengono trsformate in valori numerici e sommate ma non si può assegnare un valore numerico a una variabile stringa
- Il programma compila e assegna a s la concatenazione delle due stringhe

Crocette gennaio 2021

1. Git push
Il comando "git push" permette di:
- Caricare le modifiche locali ad alcuni file su un server condiviso
- Scaricare delle modifiche da un server condiviso
- Aggiungere delle modifiche locali ad alcuni file nella lista delle modifiche da caricare sul server
- Copiare un repository condiviso in una directory locale

2. Scopo linguaggi ad oggetti
Lo scopo principale dei linguaggi di programmazione ad oggetti e' di fornire uno strumento per

- Sviluppare programmi concisi e brevi
- Strutturare il codice in classi
- Ragionare modularmente sul codice incrementando il suo riutilizzo
- Debuggare agevolmente il codice

3. Static
Si consideri il codice seguente:

```
class A{
        public static int value = 0;
        public static void increment() {
                value++;
        }
        public static void main(String[] args) {
          A.increment();
          A obj = new A();
          obj.increment();
        }
        }
```

Quale delle seguenti affermazioni e' corretta?
- Il valore di obj.value e' 1 alla fine dell'esecuzione del metodo main
- ll valore di A.value e' 1 alla fine dell'esecuzione del metodo main
- Il valore di A.value e' 2 alla fine dell'esecuzione del metodo main
- Il codice non compila peerchè il metodo increment non è definito sull'oggetto puntato da obj

4. new
L'istruzione "new A()":
- Istanzia la classe A creando un nuovo oggetto che viene ritornato
- Istanzia la classe A, invoca il costruttore senza parametri della classe A e ritorna un riferimento al nuovo oggetto
- Invoca il metodo A()
- Accede il campo this.A

5. Protected
Un campo o metodo dichiarato protected in una classe A e' accessibile da:
- Qualsiasi classe del programma
- Solo dalle classi che si trovano nello stesso package
- Solo all'interno della classe A
- Solo dalle classi che si trovano nello stesso package, o che estendono la classe A

6. Javadoc
Javadoc e' uno standard per
- Definire commenti strutturati sulle componenti principali (come ad esempio classi, campi e metodi) di un programma Java
- Definire la firma di metodi e campi di una classe Java
- Commentare in maniera esaustiva tutte le parti del codice di un programma Java
- Compilare programmi Java

7. Extends
L'interfaccia di una classe A che estende una classe B (class A extends B) e' costituita dai metodi e campi
- Definiti nel codice della classe A
- Definiti nel codice della classe B
- Definiti nel codice della classe A e della classe B
- Definiti nel codice della classe A ma non nel codice della classe B

8. Metodo astratto
Un metodo astratto e' un metodo che
- puo' essere privato
- e' implementato
- eve essere implementato nelle sottoclassi
- e' accedibile da qualsiasi classe del programma

9. instanceof
L'istruzione obj instanceof A ritorna true se e solo se
- obj ha tipo statico A
- obj ha tipo statico A o sottotipo di A
- obj ha tipo dinamico A
- obj ha tipo dinamico A o sottotipo di A

10. Interfacce e classi
Quale delle seguenti affermazioni e' corretta?
- Una interfaccia puo' estendere al piu' un'altra interfacca
- Una classe puo' estendere molte classi
- Una classe puo' estendere al piu' un'altra classe
- Una classe puo' implementare al piu' un'interfaccia

11. Dispatch dinamico
Quale delle seguenti affermazioni e' corretta?
- Senza dispatch dinamico non possiamo avere incapsulamento (encapsulation)
- Senza dispatch dinamico non possiamo avere ereditarieta (inheritance)
- Senza dispatch dinamico non possiamo avere sottotipaggio (subtyping)
- Senza dispatch dinamico non possiamo avere polimorfismo (polymorphism)

12. hashCode
Dati due oggetti obj1 e obj2, il metodo int hashCode() definito nella classe object
- deve ritornare valori diversi se objl.equals(obj2) e' uguale a true
- deve ritornare valori diversi se objl.equals(obj2) e' uguale a false
- deve ritornare lo stesso valore se objl.equals(obj2) e' uguale a true
- deve ritornare lo stesso valore se objl.equals(obj2) e' uguale a false

13. Classe String
Si consideri il codice seguente
String s = "abc" + "def";
Quale delle seguenti affermazioni e' corretta?
- Il programma non compila in quanto non si possono sommare due stringhe costanti
- Il programma non compila in quanto le stringhe vengono trasformate in valori numerici e sommate, ma non si puo' assegnare un valore numerico a una variabile stringa
- Il programma compila e assegna a s la concatenazione delle due stringhe
- Il programma compila e assegna a s la somma dei due valori numerici ottenuti trasformando le stringhe in valori numerici

14. Try-catch
Si consideri il seguente codice:

```
try {
        foo();
        }
catch(MyException e) {
System.out.println("Error!");
}
```

L'esecuzione di tale codice, stampa a console la stringa "Error!" se e solo se
- Il metodo foo lancia un'eccezione di tipo MyException
- Il metodo foo lancia un'eccezione di tipo MyException o un sottotipo di MyException
- Il metodo foo non lancia alcuna eccezione
- Il metodo foo lancia un'eccezione qualsiasi

15. Definizione
Un'annotazione puo' essere definita attraverso:
- Un file java che definisce una classe (ad esempio, public class A {...})
- Un file java che definisce un'interfaccia (ad esempio, public interface A {...})
- Un file java che definisce un'interfaccia preceduta da @ (ad esempio, public @interface A {...})
- Un file java che definisce una classe che estende Annotation (ad esempio, public class A extends Annotation {…})

<u>Crocette giugno 2021</u>

1. Git
Il comando "git commit" permette di
- Caricare le modifiche locali ad alcuni file su un server condiviso
- Scaricare delle modifiche da un server condiviso
- Aggiungere delle modifiche locali ad alcuni file nella lista delle modifiche da caricare sul server
- Copiare un repository condiviso in una directory locale

2. Classi
(Selezionare la risposta sbagliata) Una classe
- E' composta da un insieme di campi e metodi
- Definisce uno stato (insieme di valori concreti) non modificabili
- Puo' essere istanziata in tanti oggetti
- Definisce un tipo

3. Static
Si consideri il codice seguente:

```
class A {
  public int value = 0;
  public static void increment()
  value++;
}
public static void main(String[] args){
  A.increment();
  A obj = new A();
  obj.increment();
 }
}
```

Quale delle seguenti affermazioni e' corretta?
- II valore di obj.value e' 1 alla fine dell'esecuzione del metodo main
- Il valore di A.value e' 2 alla fine dell'esecuzione del metodo main
- Il codice non compila perche' il campo value nel metodo increment non e' definito in quanto e' un campo non statico
- Il codice non compila perche' il metodo increment non e' definito sull'oggetto puntato da obj

4. Final, new, passaggio by reference
Si consideri il codice seguente:

```
class A {
   int value = 1;
   public A(int value) {
     this.value = value;
 }
}
```

Quale tra le affermazioni seguenti e' sbagliata?
- this.value=value assegna un valore al campo value definito nella classe A
- this.value=value assegna un valore al parametro value del costruttore
- classe A definisce un campo value, e un costruttore che riceve un unico parametro di tipo int
- All'inizio dell'esecuzione del costruttore, il valore di this.value e' 1

5. Firma di un metodo
La firma (signature) di un metodo e' costituita da:
Nome del metodo, tipo del valore di ritorno del metodo, numero di parametri del metodo
Nome del metodo, numero di parametri del metodo, tipi dei parametri del metodo
Nome del metodo, tipo del valore di ritorno del metodo, visibilita' del metodo
Nome del metodo, numero di parametri, tipi dei parametri del metodo, visibilita' del metodo

6. Javadoc
Javadoc e' uno standard per
- Definire commenti strutturati sulle componenti principali (come ad esempio classi, campi e metodi) di un programma Java
- Definire la firma di metodi e campi di una classe Java
- Commentare in maniera esaustiva tutte le parti del codice di un programma Java
- Compilare programmi Java

7. Extends
Il costruttore di una classe A che estende una classe B (class A extends B) che contiene solo costruttori con parametri

- Deve invocare un costruttore della superclasse B, o un altro costruttore della classe A in qualsiasi punto del codice del costruttore
- Deve invocare un costruttore della superclasse B come prima istruzione del costruttore
- Deve invocare un costruttore della superclasse B, o un altro costruttore della classe A come prima istruzione del costruttore
- Puo' invocare un costruttore della superclasse B, ma non e' necessario

8. Metodo astratto
Un metodo astratto e' un metodo che
- puo' essere privato
- e' implementato
- deve essere implementato nelle sottoclassi
- e' accedibile da qualsiasi classe del programma

9. instanceof
L'istruzione obj instanceof A ritorna true se e solo se
- obj ha tipo statico A
- obj ha tipo statico A o sottotipo di A
- obj ha tipo dinamico A
- obj ha tipo dinamico A o sottotipo di A

10. Interfacce e classi
Quale delle seguenti affermazioni e' corretta?
- Una interfaccia puo' estendere al piu' un'altra interfacca
- Una classe puo' estendere molte classi
- Una classe puo' estendere al piu' un'altra classe
- Una classe puo' implementare al piu' un'interfaccia

11. Dispatch dinamico
Quale delle seguenti affermazioni e' corretta?
- Senza dispatch dinamico non possiamo avere incapsulamento (encapsulation)
- Senza dispatch dinamico non possiamo avere ereditarieta (inheritance)
- Senza dispatch dinamico non possiamo avere sottotipaggio (subtyping)
- Senza dispatch dinamico non possiamo avere polimorfismo (polymorphism)

12. Object
Quali dei seguenti metodi non e' definito nella classe Object?
- equals
- toString
- compareTo
- hashCode

13. Classe String
Si consideri il codice seguente
String s = "abc" + "def";
Quale delle seguenti affermazioni e' corretta?
- ll programma non compila in quanto non si possono sommare due stringhe costanti
- Il programma non compila in quanto le stringhe vengono trasformate in valori numerici e sommate, ma non si puo' assegnare un valore numerico a una variabile stringa
- Il programma compila e assegna a s la concatenazione delle due stringhe
- Il programma compila e assegna a s la somma dei due valori numerici ottenuti trasformando le stringhe in valori numerici

14. Eccezioni
La relazione tra Throwable, Exception, e RuntimeException e':
- Throwable e RuntimeException estendono Exception
- Throwable estende Exception, e RuntimeException estende Exception
- Exception estende Throwable, e RuntimeException estende Exception
- Non c'e' nessuna relazione di sottotipo tra le tre classi

15. Definizione
Un'annotazione puo' essere definita attraverso:
- Un file java che definisce una classe (ad esempio, public class A {...})
- Un file java che definisce un'interfaccia (ad esempio, public interface A {...})
- Un file java che definisce un'interfaccia preceduta da @ (ad esempio, public @interface A {...})
- Un file java che definisce una classe che estende Annotation (ad esempio, public class A extends Annotation {...})