

20160218001 Nome _____ Cognome _____

A domande a risposta multipla con n scelte, la risposta corretta vale $(n-1)/n$ *difficoltàDomanda, la risposta errata $-1/n$ *difficoltà domanda, la risposta non data 0.

1. Eseguendo il frammento di codice sotto riportato, che valore assumerà la variabile "arr"?

```
int[] arr = {4, 2, 9, 13, 0};  
for(int i = 0; i < arr.length -1; i++)  
    arr[i] = arr[i + 1];
```

1. {4, 4, 2, 9, 13}
 2. {2, 2, 9, 13, 0}
 3. {0, 4, 2, 9, 13}
 4. {2, 9, 13, 0, 0}
-

2. Eseguendo il frammento di codice sotto riportato, che valore assumerà la variabile "cont"?

```
int[] arr = {22, 5, 11, 2};  
int cont = 0;  
for(int i = 0; i < arr.length; i++)  
    cont += arr[i];
```

1. 40
 2. 0
 3. 2
 4. 7
-

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

```
public interface Animale{ public String getNome(); }  
public class Cane implements Animale{ public String getNome(){return "Fufy";} }
```

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

4. Quale tra le seguenti affermazioni sul "problema della fermata" è corretta?

1. E' un problema quasi decidibile
 2. E' un problema con soluzione esponenziale
 3. E' un problema decidibile
 4. E' un problema indecidibile
-

5. Un'interfaccia deve avere?

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

6. Eseguendo il frammento di codice sotto riportato, che valore assumerà la variabile "arr"?

```
int[] arr = {4, 12, 17, 4, 8, 20, 100};  
int val = 10;  
for(int i = 0; i < arr.length; i++) {
```

```
if(arr[i] < val)
    arr[i] = val;
}
```

1. {4, 10, 10, 4, 8, 10, 10}
 2. {4, 12, 17, 4, 8, 20, 100}
 3. {10, 10, 10, 10, 10, 10, 10}
 4. {10, 12, 17, 10, 10, 20, 100}
-

7. 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);
    }
}
```

1. 32
 2. 0
 3. non esiste un costruttore della classe che accetti in input un parametro
 4. 4
-

8. Eseguendo il frammento di codice sotto riportato, che valore assumerà la variabile "var"?

```
int[] arr = {6, 4, 2, 1, 3, 5};
int val = 0;
for(int i = 0; i < arr.length; i++) {
    if(arr[i] < arr.length / 2)
        val -= arr[i];
    else
        val += arr[i];
}
```

1. 15
 2. 0
 3. 3
 4. 6
-

9. Quale tra le seguenti affermazioni sulla "ricorsione" è corretta?

1. Ogni metodo ricorsivo può essere reso non ricorsivo
 2. In Java non è possibile scrivere metodi ricorsivi
 3. Un metodo ricorsivo è generalmente più efficiente del corrispondente metodo iterativo
 4. La ricorsione ha un potere espressivo superiore all'iterazione
-

10. Cosa permette di fare il codice riportato di seguito?

```
BufferedReader input = new BufferedReader(new FileReader("dati.txt"));
String str;
while( (str = input.readLine()) != null)
    System.out.println(str);
```

1. Permette di aprire il file "dati.txt" in scrittura e scrivere una riga vuota
 2. Permette di aprire il file "dati.txt" in lettura e di leggere la prima riga
 3. Permette di aprire il file "dati.txt" in lettura e di contare quanti caratteri "a capo" ci sono
 4. Permette di aprire il file "dati.txt" in lettura e di leggere l'intero file
-

11. Che valore restituisce la chiamata mioMetodo("qwerty"), se il metodo è implementato come segue?

```
public int mioMetodo(String str) {  
    if(str.equals(""))  
        return 0;  
    else  
        return 2 + mioMetodo(str.substring(1, str.length()));  
}
```

1. 0
 2. 1
 3. 12
 4. Il metodo entra in un loop di chiamate infinito
-

12. Eseguendo il codice sotto riportato, che valore assumerà l'attributo anni dell'istanza persona?

```
public class Persona{  
    int anni;  
    public Persona(int anni){this.anni=anni;}  
    public Persona(){this.anni=12;}  
    public static void main(String[] args){  
        Persona persona = new Persona(23);  
    }  
}
```

1. 23
 2. 0
 3. non esiste un costruttore della classe che accetti in input un parametro
 4. 12
-

13. Come si definisce una classe che gestisca eccezioni personalizzate?

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

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

```
String testo = "sono uno studente che sta facendo il test";  
int indice = testo.indexOf("un");
```

1. 2
 2. 5
 3. 7
 4. 4
-

15. Quale tra questi tipi di dato primitivi è corretto per memorizzare il carattere 'C'?

1. double
 2. boolean
 3. char
 4. float
-

16. Eseguendo il frammento di codice sotto riportato, che valore assumerà la variabile "cont"?

```
int i = 10;  
int cont = 8;  
while(i < cont)
```

```
{
  cont --;
  i++;
}
```

1. il ciclo è infinito
 2. 10
 3. 8
 4. 9
-

17. Eseguendo il frammento di codice sotto riportato, che valore assumerà la variabile "nome"?

```
int eta=2;
String nome="";
switch(eta) {
  case 0:
    nome="Stefano";
    break;
  case 1:
    nome="Ruggero";
    break;
  case 2:
    nome="Giacomo";
    break;
  case 3:
    nome="Pietro";
    break;
}
```

1. "Stefano"
 2. "Pietro"
 3. "Ruggero"
 4. "Giacomo"
-

18. Eseguendo il frammento di codice sotto riportato, che valore assumerà la variabile "arr"?

```
int[] arr = new int[5];
for(int i = 0; i < arr.length; i++)
{
  if(i % 2 == 0)
    arr[i] = 0;
  else
    arr[i] = i;
}
```

1. {0, 0, 0, 3, 0}
 2. {0, 2, 0, 4, 0}
 3. {1, 0, 3, 0, 5}
 4. {0, 1, 0, 3, 0}
-

19. Quale tra le seguenti affermazioni sulla "macchina di Von Neumann" è corretta?

1. E' l'architettura alla base della Pascalina di Blaise Pascal
 2. La memoria contiene dati e programmi
 3. E' una macchina meccanica in grado di eseguire solo addizioni
 4. E' una macchina meccanica in grado di eseguire le quattro principali operazioni aritmetiche
-

20. Eseguendo il frammento di codice sotto riportato, che valore assumerà la variabile "var"?

```
int[] arr = {6, 4, 2, 1, 3, 5};
```

```
int val = 0;
for(int i = 0; i < arr.length; i++) {
    if(i < arr.length / 2)
        val += arr[i];
    else
        val -= arr[i];
}
```

1. 6
 2. 15
 3. 0
 4. 3
-

21. Eseguendo il codice sotto riportato, che valore assumerà la variabile n?

```
int n;
double d = 1.98;
n = (int)d;
```

1. 1
 2. 2
 3. 1.98
 4. 0
-

22. Eseguendo il frammento di codice sotto riportato, cosa verrà stampato a video?

```
int val = 10;
do {
    int r = val % 2;
    val = val / 2;
    System.out.print(r);
} while(val >= 1);
```

1. 1010
 2. 0101
 3. 1001
 4. il ciclo è infinito
-

23. Eseguendo il frammento di codice sotto riportato, cosa verrà stampato a video?

```
int i = 2;
do {
    System.out.print(i + " ");
    i ++;
} while (i <= 9);
```

1. 2 3 4 5 6 7 8
 2. 2
 3. il ciclo è infinito
 4. 2 3 4 5 6 7 8 9
-

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

```
public class Animale{int nZampe; int eta;}
public class Cane extends Animale{String nome;}
```

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

25. Eseguendo il codice sotto riportato, che valore assumeranno le variabili x e y?

```
int x = 3;
int y = 12;
int z = x;
x = y;
y = z;
```

1. x=12, y=12
 2. x=3, y=12
 3. x=3, y=3
 4. x=12, y=3
-

26. Quale è il costrutto per implementare un'interfaccia?

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

27. Cosa è il byte-code?

1. Codice di crittografia per le transazioni bancarie
 2. Linguaggio intermedio eseguito dalla macchina virtuale Java
 3. Sequenza di 8 bit
 4. Codifica in byte successiva all'ASCII
-

28. In quale tra questi frammenti di codici la condizione del costrutto "if" non risulta vera?

1.

```
int numero =26;
if(numero <34){}
```
 2.

```
int numero=54;
if(numero<45){}
```
 3.

```
int numero=6;
if(numero ==6){}
```
 4.

```
int numero =22;
if(numero >=22){}
```
-

29. Quale tra le seguenti affermazioni sulla "distruzione di oggetti in Java" è corretta?

1. I metodi distruttori in Java devono avere lo stesso nome della classe preceduto da una tilde (~)
 2. In Java tutti gli oggetti rimangono sempre in vita fino alla fine dell'esecuzione del programma
 3. Java fa uso di distruttori espliciti
 4. Java fa uso del garbage collector per riciclare memoria non più referenziata
-

30. Quale è l'effetto del seguente codice?

```
double x = 5.0;
int n;
n = x;
```

1. Converte implicitamente 5 in 5.0
2. Converte implicitamente 5.0 in 5
3. Da errore di compilazione

4. Effettua un troncamento del valore 5 in 0

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

```
String testo = "La nebbia agli irti colli";  
int numero = testo.length();
```

- 1. 30
 - 2. 20
 - 3. 25
 - 4. -1
-

32. Eseguendo il frammento di codice sotto riportato, che valore assumerà la variabile "cont"?

```
int cont = 0;  
for(int i = 0; i < 15; i++)  
{  
    cont++;  
    i++;  
}
```

- 1. 8
 - 2. 15
 - 3. 0
 - 4. 7
-

33. Qual'è il codice corretto per realizzare il costrutto try-catch?

- 1. try {} catch {}
 - 2. try {} catch (Exception e){}
 - 3. try () catch ()
 - 4. try () throws ()
-

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

```
public class Cane{ int nZampe; }
```

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

35. 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 44;} }  
public class SottoClasse extends SuperClasse{ public int getNumero(){return 32;} }
```

- 1. 76
- 2. 32
- 3. 44
- 4. non è possibile effettuare la chiamata al metodo