

parte teorica 08.04.2022

Test e valutazione - Anteprima di stampa

parte teorica 08.04.2022

Data: Tue May 3 12:18:15 2022 Punteggi massimi: 63

Question 1 - Dynamic Binding Java (2) [ID: 304528]

Date le seguenti dichiarazioni:

```
1  class Persona {
2      void setAge(int l) {
3          System.out.println("P");
4      }
5  }
6
7  class Studente extends Persona {
8      void setAge(int l) {
9          System.out.println("S");
10     }
11 }
12
13 class Anziano extends Persona {
14     void setAge(long l) {
15         System.out.println("A");
16     }
17 }
18
19 ...
20 Object op = new Persona();
21 Persona pp = new Persona();
22 Persona ps = new Studente();
23 Persona pa = new Anziano();
24 Anziano aa = new Anziano();
25 int age = 30;
26 long ageL = 200;
```

Qual è l'output prodotto dalle sequenti istruzioni (errore se pensi ci sia un errore)?

- op.setAge(age); errore (2 Punti)
- pp.setAge(age); P (1 Punto)
- ps.setAge(age); S (1 Punto)
- pa.setAge(age); P (4 Punti)
- pa.setAge(ageL); errore (1 Punto)
- aa.setAge(age); P (4 Punti)
- aa.setAge(ageL); A (2 Punti)

ERRORE
P
S
P
ERRORE
P
A

Question 2 - Return result address [ID: 304531]

Che cos'è il return result-address?

- ☒ un campo contenente l'indirizzo dove salvare il risultato della funzione (1 Punto)
- ☐ un campo contenente l'istruzione return della funzione (-1 Punti)
- ☐ un campo contenente l'indirizzo della prima istruzione da eseguire quando la funzione termina (-1 Punti)
- ☐ un campo contenente l'indirizzo della funzione chiamata "return" (-1 Punti)

Question 3 - C++ virtual destructors ed ereditarietà (1) [ID: 304532]

Date le seguenti classi:

```

1  struct A {
2      A() { cout << "A+"; }
3      virtual ~A() { cout << "A-"; }
4  };
5
6  struct B {
7      B() { cout << "B+"; }
8      ~B() { cout << "B-"; }
9  };
10
11 struct C {
12     C() { cout << "C+"; }
13     ~C() { cout << "C-"; }
14 };
15
16 struct X : A, B, protected C {
17     X() { cout << "X+"; }
18     ~X() { cout << "X-"; }
19 };

```

Scrivi l'output delle seguenti istruzioni (ERR se pensi che ci sia un errore):

A* a = new A; delete a; A+A- (1 Punto)

A+

A* a = new X; delete a; A+B+C+X+X-C-B-A- (1 Punto)

B* b = new B; delete b; B+B- (1 Punto)

B* b = new X; delete b; A+B+C+X+B- (1 Punto)

C* c = new C; delete c; C+C- (1 Punto)

C* c = new X; delete c; ERR (1 Punto)

X* x = new X; delete x; A+B+C+X+X-C-B-A- (1 Punto)

Question 4 - Ridefinizione di un metodo Java (3) MR [ID: 304535]

Cosa accade se in una sottoclasse B introduco un metodo m fatto nei seguenti modi? Scegli "errore" se ho un errore in compilazione "overload" se faccio overload o "override" se sto facendo override.

Data la seguente classe:

```
class A {
```

```
protected Object foo(int x){return new Integer(x);}

}

protected Object foo(int x){return null;}  override (1 Punto)
Object foo(int x){return null;}  errore (1 Punto)
private Object foo(int x){return null;}  errore (1 Punto)
protected Object foo(float x){return null;}  overload (1 Punto)
protected String foo(int x){return null;}  override (1 Punto)
```

OVERRIDE
ERRORE
ERRORE
OVERLOAD
OVERRIDE

Question 5 - Java generics - wildcard (2) MR [ID: 304536]

```
1  ArrayList<String> li = new ArrayList<String>();
2
3  li.add("1"); li.add("2"); li.add("3");
4
5  stampaLista(li);
```

Dire, per ognuna delle seguenti implementazioni di "stampaLista" (NOTA: cambia solo il parametro in input), cosa stampano in output (se stampano correttamente la lista oppure se danno errore in compilazione):

```
void stampaLista(List<Integer> list) { for (Object elem : list) System.out.print(elem + " "); }
```

errore ERRORE

(1 Punto)

```
void stampaLista(ArrayList<Object> list) { for (Object elem : list) System.out.print(elem + " "); }
```

errore ERRORE

(1 Punto)

```
void stampaLista(List<String> list) { for (Object elem : list) System.out.print(elem + " "); }
```

1 2 3 1 2 3

(1 Punto)

```
void stampaLista(ArrayList<?> list) { for (Object elem : list) System.out.print(elem + " "); }
```

(1 Punto)

void stampaLista(List<T> list) { for (Object elem : list) System.out.print(elem + " "); } errore (1 Punto)

void stampaLista(List<? super String> list) { for (Object elem : list) System.out.print(elem + " "); } 1 2 3 (1 Punto)

void stampaLista(List<? super Object> list) { for (Object elem : list) System.out.print(elem + " "); } errore (1 Punto)

void stampaLista(List<? extends String> list) { for (Object elem : list) System.out.print(elem + " "); } 1 2 3 (1 Punto)

void stampaLista(List<? implements Comparable> list) { for (Object elem : list) System.out.print(elem + " "); } errore (1 Punto)

Question 6 - Java generics - tipi (4) MR [ID: 304537]

```
1 class Shape { /* ... */ }
2 class Circle extends Shape { /* ... */ }
3 class Rectangle extends Shape { /* ... */ }
```

Dire, per ognuno dei seguenti pezzi di programma, se è corretto o se presenta errori. Nel caso di errore, indicare la PRIMA linea di codice che contiene l'errore:

// -- Codice 1:

```
class Node<T> { /* ... */ }
Node<Circle> nc = new Node<Circle>();
Node<Shape> ns = nc; errore istruzione 3 (1 Punto)
```

// -- Codice 2:

```
class Node<?> { /* ... */ }
Node<Circle> nc = new Node<Circle>();
Node<Circle> ns = nc; errore istruzione 1 (2 Punti)
```

// -- Codice 3:

```
class Node<T> { /* ... */ }
Node<Circle> nc = new Node<Shape>();
Node<Circle> ns = nc; errore istruzione 2 (1 Punto)
```

// -- Codice 4:

```
class Node<T> { /* ... */ }
Node<Shape> nc = new Node<Circle>();
Node<Shape> ns = nc; errore istruzione 2 (1 Punto)
```

// -- Codice 5:

```
class Node<T> { /* ... */ }
```

```
Node<?> nc = new Node<?>();
```

```
Node<Shape> ns = nc; errore istruzione 2 (1 Punto)
```

2

// -- Codice 6:

```
class Node<T> { /* ... */ }
```

```
Node<Circle> nc = new Node<>(); corretto (1 Punto)
```

,

On

// -- Codice 7:

```
class Node<PIPPPO> { /* ... */ }
```

```
new Node<>(); corretto (1 Punto)
```

On

// -- Codice 8:

```
Shape s = null;
```

```
Circle c = s; errore istruzione 2 (1 Punto)
```

2

// -- Codice 9:

```
class Node<K> { /* ... */ }
```

```
Node<Circle> nc = new Node<>();
```

```
Node<Circle> ns = nc; corretto (1 Punto)
```

On

// -- Codice 10:

```
class MyList extends ArrayList<Rectangle> { /* ... */ }
```

```
MyList nc = new MyList();
```

```
ArrayList<Rectangle> ns = nc; corretto (1 Punto)
```

On

Question 7 - Overriding/Overloading (0) [ID: 304538]

Dato il seguente codice

```
1  class Value {}
2  class SmallValue extends Value {}
3
4  class Elaboratore {
5      Value getVal() {
6          return new Value();
7      }
8  }
9
10 class Phone extends Elaboratore {
11     SmallValue getVal() {
12         return new SmallValue();
13     }
14 }
```

Quali di queste affermazioni sono giuste?

- ☒ Phone è una sottoclasse di Elaboratore (Selezionato = 1 Punto, Non selezionato = 0 Punti)
- ☒ Phone fa overriding del metodo getVal di Elaboratore (Selezionato = 1 Punto, Non selezionato = 0 Punti)
- ☐ Phone fa overloading del metodo getVal di Elaboratore (Selezionato = 0 Punti, Non selezionato = 1 Punto)
- ☐ Phone contiene un errore: non può definire getVal in questo modo! (Selezionato = 0 Punti, Non selezionato = 1 Punto)

Question 8 - Ridefinizione di metodi con classi (2) MQ [ID: 304539]

```
1  class Animale{
2      public Animale get(){return null;}
3  }
4  class Cane extends Animale {
5      private Animale get(){return null;}
6  }
7  class Gatto extends Animale {
8      public Animale get(){return null;}
9  }
10 class Topo extends Animale {
11     public Gatto get(){return null;}
12 }
```

[Download](#)

Quali di questi metodi sono ridefiniti in modo sbagliato (errore in compilazione)?

- ☐ il metodo get di Topo (Selezionato = -1 Punto, Non selezionato = 1 Punto)
- ☒ il metodo get di Cane (Selezionato = 1 Punto, Non selezionato = -1 Punto)
- ☐ il metodo get di Gatto (Selezionato = -1 Punto, Non selezionato = 1 Punto)

Question 9 - Passaggio di array in C (0) [ID: 304540]

Data la seguente funzione

```

1 void f(int a[]){
2     printf("%d\n",sizeof(a));
3     a = a +1;
4 }
5
6
7 int main(void) {
8     int p[] = {10,20,30};
9     printf("%d\n",sizeof(p));
10    f(p);
11    printf("%d\n",*p);
12    return EXIT_SUCCESS;
13 }

```

Qual'è l'output prodotto dalle seguenti istruzioni (in ordine di esecuzione)? Se pensi contenga un errore, scrivi errore

Assumi che un puntatore vale 4 byte come anche un intero (32 bit).

nel main:

printf("%d\n",sizeof(p)); 12 (1 Punto)

12

in f:

printf("%d\n",sizeof(a)) 4 (4 Punti)

4

nel main di nuovo

printf("%d\n",*p); 10 (2 Punti)

10

Question 10 - record di attivazione (2) [ID: 304543]

Considera il seguente codice:

```

int f(int &y, int *x) {
    cout << x[0] << x[1] << y << " ";
    if (*x == y) {
        int q = *(x + 1) * ++y;
        return f(q, x + 2);
    }
    return ++y + *x;
}

int main() {
    int a[] = { 3, 2, 1, 0 };
    int &b = a[0];
    int d = f(b, a);
    cout << a[0] << a[1] << a[2] << a[3] << b << " "<< d << endl;
    return 0;
}

```

Disegna il record di attivazione e consegna il foglio. Tu non usare ottimizzazioni, ma si potrebbero applicare in questo caso? Come risulterebbe il RA?

Quale è l'output del programma?

1. 323 108 42104 10 (1 Punto)

