Questionario di auto-valutazione del Laboratorio 8

Corso: Fondamenti di Informatica 1 (gruppo 8-9)

Corsi di laurea: Area dell'Informazione

Domanda n. 1

Argomento: Classi

Peso: 1

Pubblica: N

Domanda: In Java una classe:

Risposte:

1. in assenza di estensione esplicita di altra classe, estende la classe Object

Punteggio: 1

2. puo' estendere piu' di una classe

Punteggio: 0

3. non puo' realizzare (implement) piu' di una interfaccia

Punteggio: 0

4. nessuna delle affermazioni precedenti e' corretta

Punteggio: 0

Spiegazione:

In Java una classe estende la classe Object in assenza di esplicita estensione; puo' estendere esplicitamente una sola classe e puo' realizzare piu' interfacce.

Domanda n. 2

Argomento: Classi

Peso: 1

Pubblica: N

Domanda: Sia D una classe che estende la classe C. Il seguente frammento di codice in Java:

```
public class Test
{ public static void main(String[] args)
  { Object x = new D();
    C y = (D) x;
  }
}
```

Risposte:

1. e' corretto, dal momento che nell'enunciato $C y = (D) x$; si usa il forzamento esplicito

Punteggio: 1

2. e' corretto, ma sarebbe corretto anche senza il forzamento esplicito essendo D una sottoclasse di C

Punteggio: 0

3. non e' corretto: in esecuzione viene generata l'eccezione `java.lang.ClassCastException`

Punteggio: 0

4. non viene compilato a causa di errore di tipo nell'istruzione `Object x = new D();`

Punteggio: 0

Spiegazione:

E' corretto. Infatti l'espressione

```
Object x = new D();
```

crea un esemplare della classe D e assegna il riferimento alla variabile riferimento x che e' di tipo Object. La conversione automatica fra le classi D e Object e' automatica perche' la classe Object e' una super-classe della classe D.

L'espressione

```
C y = (D) x;
```

forza il riferimento x al tipo D. Il forzamento non genera in esecuzione l'eccezione `ClassCastException` perche' l'esemplare a cui si riferisce x e' stato creato di tipo D. L'assegnazione alla variabile C y e' corretta perche' la classe C e' superclasse di D.

Domanda n. 3

Argomento: Classi

Peso: 1

Pubblica: N

Domanda: Sia D una classe che estende la classe C. Il seguente frammento di codice in Java:

```
public class Test
{ public static void main(String[] args)
  { Object x = new C();
    D y = (C) x;
  }
}
```

Risposte:

1. e' corretto, dal momento che nell'enunciato $D y = (C) x$; si usa il forzamento esplicito

Punteggio: 0

2. e' corretto, ma sarebbe corretto anche senza il forzamento esplicito essendo D una sottoclasse di C

Punteggio: 0

3. non e' corretto: in esecuzione viene generata l'eccezione `java.lang.ClassCastException`

Punteggio: 0

4. non viene compilato a causa di errore di tipo nell'istruzione $D y = (C) x$;

Punteggio: 1

Spiegazione:

Non viene compilata a causa di errore di tipo nell'istruzione

```
D y = (C) x;
```

infatti D e' sottoclasse di C.

Domanda n. 4

Argomento: Classi

Peso: 1

Pubblica: N

Domanda: Una sottoclasse della classe X e' una classe che

Risposte:

1. eredita tutti i metodi di X

Punteggio: 1

2. realizza i metodi di cui X descrive soltanto l'intestazione

Punteggio: 0

3. realizza i metodi protected di X

Punteggio: 0

4. nessuna delle affermazioni precedenti e' corretta

Punteggio: 0

Spiegazione:

Eredita tutti i metodi di X.

Domanda n. 5

Argomento: Classi

Peso: 1

Pubblica: N

Domanda: Se D e' una classe derivata dalla classe B:

Risposte:

1. nessun metodo della classe D puo' avere il nome di un metodo della classe B

Punteggio: 0

2. tutti i metodi della classe D devono avere il nome di un metodo della classe B

Punteggio: 0

3. i metodi della classe D che hanno lo stesso nome, lo stesso numero e tipo dei parametri di quelli di B sovrascrivono i metodi di B

Punteggio: 1

4. nessuna delle precedenti affermazioni e' corretta

Punteggio: 0

Spiegazione:

I metodi della classe D che hanno lo stesso nome, lo stesso numero e tipo dei parametri di quelli di B sovrascrivono i metodi di B. Si veda la sintassi di Java.

Domanda n. 6

Argomento: Classi

Peso: 1

Pubblica: N

Domanda: Qual e' l'andamento asintotico della complessita' temporale del seguente metodo, detto n il numero di elementi dell'array passato come parametro?

```
private static void rovescia(int[] a)
{
    int i = 0;
    while (i <= a.length/2)
    {
        int tmp = a[i];
        a[i] = a[a.length - 1 - i];
        a[a.length - 1 - i] = tmp;
        i++;
    }
}
```

Risposte:

1. $O(1)$

Punteggio: 0

2. $O(\log n)$

Punteggio: 0

3. $O(n)$

Punteggio: 1

4. $O(n \log n)$

Punteggio: 0

5. $O(n^2)$ dove n^2 significa n al quadrato

Punteggio: 0

Spiegazione:

Il ciclo esegue $a.length/2$ iterazioni, pari a $n/2$. La complessità temporale è quindi $O(n)$.

Domanda n. 7

Argomento: Algoritmi e complessità temporale

Peso: 1

Pubblica: N

Domanda: Se per ordinare un insieme di 1.000 chiavi con l'algoritmo di ordinamento per fusione occorrono 10 secondi, quanti secondi saranno necessari per ordinare 10.000 chiavi?

Risposte:

1. meno di 100 secondi

Punteggio: 0

2. più di 100 ma meno di 500 secondi

Punteggio: 1

3. più di 500 ma meno di 1000 secondi

Punteggio: 0

4. più di 1000 secondi

Punteggio: 0

Spiegazione:

Sono necessary circa 133 s. Infatti, essendo l'andamento asintotico dell'algoritmo mergesort pari a $n \log n$:

$$\tau_2 \approx \tau_1 \frac{n_2 \log n_2}{n_1 \log n_1}$$

Domanda n. 8

Argomento: Algoritmi e complessita' temporale

Peso: 1

Pubblica: N

Domanda: Sia dato un elenco di 1048 nomi non ordinati alfabeticamente. Se per ordinare l'elenco si impiega al meglio 1 ms, quanti millisecondi sono necessari per ordinare un elenco non ordinato di 1.000.000 elementi?

Risposte:

1. meno di 10 ms

Punteggio: 0

2. piu' di 10 ms ma meno di 100 ms

Punteggio: 0

3. piu' di 100 ms ma meno di 1 secondo

Punteggio: 0

4. piu' di 1 secondo ma meno di 10 secondi

Punteggio: 1

5. piu' di 10 secondi

Punteggio: 1

Spiegazione:

Si impiega piu' di un secondo, ma meno di 10 s. Infatti, ordinare al meglio significa ordinare con un algoritmo di andamento asintotico pari a $O(n \log n)$.

Domanda n. 9

Argomento: Classi

Peso: 1

Pubblica: N

Domanda: La compilazione della seguente classe, che realizza l'interfaccia parametrica `java.lang.Comparable`,

```
public class Z implements Comparable
{
    private int n;

    public Z(int k) { n = k; }
    public int compareTo(Z z)
    {
        return n - z.n;
    }
}
```

Risposte:

1. non genera errori

Punteggio: 0

2. genera un errore nell'enunciato `return n - z.n;`

Punteggio: 0

3. genera un errore perché la classe non realizza il metodo `int compareTo()` dell'interfaccia `java.lang.Comparable`

Punteggio: 1

4. nessuna delle precedenti affermazioni è corretta

Punteggio: 0

Spiegazione:

Genera un errore in compilazione perché non è definito nella classe il metodo prescritto dall'interfaccia `Comparable` che ha la seguente intestazione `public int compareTo(Object obj)`.

Il metodo `public int compareTo(Z z)` non è riconosciuto come metodo prescritto dall'interfaccia `Comparable`.

Domanda n. 10

Argomento: Classi

Peso: 1

Pubblica: N

Domanda: La compilazione della seguente classe, che realizza l'interfaccia parametrica `java.lang.Comparable<T>`,

```
public class Z implements Comparable<Z>
{
    private int n;

    public Z(int k) { n = k; }
    public int compareTo(Z z)
    {
        return n - z.n;
    }
}
```

Risposte:

1. non genera errori

Punteggio: 1

2. genera un errore nell'enunciato `return n - z.n;`

Punteggio: 0

3. genera un errore perché la classe non realizza il metodo `compareTo` dell'interfaccia `java.lang.Comparable`

Punteggio: 0

4. nessuna delle precedenti affermazioni è corretta

Punteggio: 0

Spiegazione:

La classe non genera errori in compilazione.

Domanda n. 11

Argomento: Algoritmi e complessita' temporale

Peso: 1

Pubblica: N

Domanda: La funzione $f(n)$ e' $O(g(n))$ se:

Risposte:

1. esistono due costanti positive c e n_0 tali $0 \leq f(n) \leq c g(n)$, per $n \geq n_0$

Punteggio: 1

2. esistono due costanti positive c e n_0 tali $0 \leq g(n) \leq c f(n)$, per $n \geq n_0$

Punteggio: 0

3. esiste una costante positiva c tale che $0 \leq f(n) \leq c g(n)$, per ogni $n > 0$

Punteggio: 0

4. nessuna delle precedenti affermazioni e' corretta

Punteggio: 0

Spiegazione:

Si veda la definizione di $O(g(n))$.

Domanda n. 12

Argomento: Algoritmi e complessita' temporale

Peso: 1

Pubblica: N

Domanda: Dato un programma con complessita' $O(n^2)$, se il numero dei dati di ingresso raddoppia, il tempo di esecuzione

Risposte:

1. raddoppia

Punteggio: 0

2. triplica

Punteggio: 0

3. quadruplica

Punteggio: 1

4. si moltiplica per 8

Punteggio: 0

Spiegazione:

Il tempo di esecuzione quadruplica.