

Prova scritta – IV appello, 12 luglio 2006

Cognome

Nome

Matricola

1

Stabilire qual'è l'output del seguente codice Java, supponendo che sia stata dichiarata la variabile `int i`:

<pre>for (i = 1; i < 10; i++) { if (i%3 == 0) break; System.out.println (i); } System.out.println (i);</pre>	
<pre>for (i = 1; i < 10; i++) { if (i%3 == 0) continue; System.out.println (i); } System.out.println (i);</pre>	
<pre>for (i = 1; i < 10; i++) { if (i%3 == 0) System.exit(0); System.out.println (i); } System.out.println (i);</pre>	

2

Supponete sia definita una classe `BarcaVela` derivata dalla classe `Barca` e supponete che quest'ultima classe abbia definito i metodi pubblici `velocitàNodi()` e `accendiMotori()`.

Un oggetto di classe `BarcaVela` risponde ai metodi `velocitàNodi()` e `accendiMotori()`? SI NO

Questi metodi dovranno eseguire le stesse azioni sia per gli oggetti di classe `BarcaVela` sia per gli oggetti di classe `Barca`, sempre e comunque? SI NO

Come si chiama in Java il concetto di specializzazione dei metodi?

Se per la classe `BarcaVela` è stato definito il metodo `alzareRanda()`, questo può essere invocato su un oggetto di classe `Barca`? SI NO

3

Data la classe seguente con i costruttori elencati:

<pre>class Dipendente { private String nome;</pre>	<pre>Dipendente(){ nome = new String(); } Dipendente(String unNome) { nome = unNome; } }</pre>
--	--

E' possibile cambiare il valore del membro `nome` di un oggetto già creato? SI NO
Come? (scrivere le istruzioni necessarie)

Per estendere la classe con una specializzazione, introduciamo la sottoclasse `Impiegato`, completare quindi il codice seguente opportunamente:

<pre>class Impiegato extends Dipendente { private int settore; // 1, 2, 3 ad esempio Impiegato() { settore = 1; // valore di default } }</pre>	<pre>Impiegato (String unNome, int unSettore) { settore = unSettore; } }</pre>
---	---

Supponendo di aver eseguito l'istruzione: `Impiegato imp = new Impiegato();`

E' possibile accedere al campo `nome` della superclasse? SI NO

Per accedere al campo `nome` è corretta l'espressione: `imp.nome` ? SI NO Esistono altri modi? SI NO

Definire il codice di possibili alternative, per accedere al campo:

In generale, in una sottoclasse è possibile accedere direttamente ad un campo privato di un'istanza della superclasse? SI NO

In una sottoclasse è possibile usare un metodo privato della superclasse? SI NO

4

Scrivere il metodo `init` che inizializza con numeri dispari un array di interi di dimensione `MAX`, passato come parametro. L'array inizializzato andrà riportato all'ambiente chiamante. A scelta si usi un ciclo `for` o un ciclo `while`:

```
public int[] init(int[] a) {
```

```
}
```

5

Scrivere un metodo per rimuovere un elemento in posizione `data` da un array che verrà riportato all'ambiente chiamante modificato. Sia la posizione dell'elemento da rimuovere sia l'array sono passati come parametri. Per rimuovere l'elemento si deve procedere ad eseguire l'avanzamento di una posizione di tutti i valori dell'array dalla posizione `data` alla fine. Si supponga che l'array sia un array di interi.

```
int[] rimuovi(int i, int[] a) {
```

```
}
```

6

Utilizzando la classe `StringTokenizer` implementare un ciclo `for` necessario per inizializzare l'array di stringhe `frase`. Ad esempio: se in input è data la frase "questo è un esame" allora l'array `frase` dovrà essere inizializzato a `[questo, è, un, esame]`.

```
System.out.print("input una riga di testo: ");
String riga = Keyboard.readString();
StringTokenizer processore_di_riga = new           ;

String [] frase = new                           ;

for
```

7

Scrivere il metodo `int conta()` che conta i numeri pari contenuti in un array membro d'istanza della classe `Numeri`, la cui lunghezza è stabilita da una costante `MAX` e che viene inizializzato con numeri pseudo casuali compresi tra 1 e 10 (estremi inclusi).

```
class Numeri {
    final int MAX = 20;
    private int [] array_numeri;
```

8

Dire se è corretto il seguente ciclo `for`, dove `v` è un oggetto di classe `Vector`:

```
for (Enumeration e = v.elements(); e.hasMoreElements();) {
    System.out.println(e.nextElement());
}
```

Specificare il perché:

Dire quale metodo viene implicitamente invocato quando si passa l'argomento `e.nextElement()` al metodo `println`:

9

Implementare l'interfaccia `Enumeration` del pacchetto `java.util` per la classe `EnumerationDemo`:

```
public class EnumerationDemo implements java.util.Enumeration {
    private String[] mazzo =
        {"rosa", "dalia", "zinia", "margherita", "narciso"};
    private int indice = 0;
    public boolean hasMoreElements() {
```

```
        public Object nextElement() {
```

10

Disegnate lo stato dello stack che viene utilizzato per valutare l'espressione aritmetica in notazione postfissa

1 5 + 2 * 3 1 + /
 ^

quando si sia arrivati a scandire l'espressione fino al token 3.

