Corso di laurea in Comunicazione digitale — Fondamenti di architettura e programmazione

AA 2008/09 — Prof. M.A. Alberti

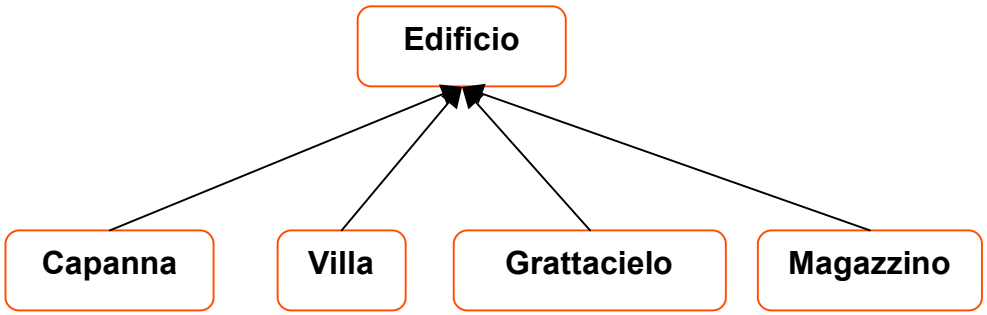
II Appello - 23 febbraio 2009

Cognome **Nome**

Matricola

1 (8 punti)

Definite la gerarchia di classi per tipi differenti di edifici, come illustrato nella figura:



Nella super-classe dichiarate i campi **`superficie`**, **`anni`**, **`indiceLocazione`**. Quest'ultimo campo archivia un numero decimale che deve essere usato per calcolare il valore dell'edificio in funzione della sua locazione.

Estendete la classe base per definire le sotto-classi indicate, oltre che ad un'altra classe di vostra scelta. Per ciascuna sottoclasse, includete almeno un nuovo campo d'istanza che sia significativo e diverso da quelli della classe base.

Per ciascuna sotto-classe definite il metodo **`valutazione()`** che ritorna il valore dell'edificio e che usa in modo logico e ragionevole i campi **`superficie`** e **`anni`**, che indicano rispettivamente la superficie dell'edificio e la sua età, il campo **`indiceLocazione`** e quelli addizionali che avete definito voi.

Lo scopo dell'esercizio consiste nel ricercare un disegno object-oriented accurato; i requisiti per le classi sono specificati in modo generico. Non occorre scrivere codice ma pensare a problematiche quali: cosa definire **`private`**, ha senso definire una classe **`abstract`**, cosa ereditano le sottoclassi etc.

---

2 (3 punti)

Assumete che le variabili **`a`** e **`b`** siano definite **`boolean`**. Scegliete tra le situazioni seguenti quella che descrive il fatto che l'espressione **`!(a && b) && (a || b)`** venga valutata vera.

a. Sempre

b. Mai

c. Quando sia **`a`** che **`b`** sono **`true`**

d. Quando nè **`a`** nè **`b`** sono **`true`**

e. **Solo quando una sola delle variabili, `a` o `b`, è `true`**

a	b	a && b	!(a && b)	a || b	!(a&&b)&&(a||b)
F	F	F	T	F	F
F	T	F	T	T	T
T	F	F	T	T	T
T	T	T	F	T	F

3 (3 punti)

Lo spezzone di codice seguente dovrebbe dire se la variabile **int num** è pari o dispari. Non funziona però. Come può essere corretta?

```
Switch (num % 2) {                                  // linea 1
    case 1: System.out.println("numero dispari"); // linea 2
    case 0: System.out.println("numero pari");    // linea 3
}
```

a. **Aggiungere uno statement `break` alla fine della linea 2.**

b. Aggiungere uno statement **`break`** alla fine della linea 3.

c. Aggiungere uno statement continue alla fine della linea 2.

d. Scambiare la linea 2 con la linea 3.

e. Cambiare l'operatore `%` con l'operatore **`/`** nella linea 1.

4 (3 punti)

Considerate i due frammenti di codice seguenti (assumete **x** una variabile di tipo **int**):

```
while ( x>0 ) { x-- }                  // frammento 1
System.out.println("x = " + x);

do ( x-- ) while ( x>0 );              // frammento 2
System.out.println("x = " + x);
```

In quale tra le seguenti circostanze l'output dei due frammenti è diverso?

I. **x è 0 prima che il segmento sia eseguito**

II. **x** è maggiore di **0** prima che il segmento sia eseguito

III. **x è minore di 0 prima che il segmento sia eseguito**

Risposta:

a. solo I b. solo II c. solo III d. I e II e. **I e III**

5 (3 punti)

Tra le seguenti affermazioni, dire qual è la **migliore** ragione per dichiarare private piuttosto che public tutti i campi di una classe?

a. Se i campi di una classe sono **`private`**, i metodi sono dichiarati **`public`**.

b. **Se i campi di una classe sono `private`, i loro identificatori possono essere cambiati senza richiedere cambiamenti nel codice della classe che li usa.**

c. Se i campi di una classe sono **`private`**, il compilatore può generare codice più efficiente per i metodi della classe.

d. Se i campi di una classe sono **`public`**, ci sarebbe una scelta limitata d'identificatori.

e. Se i campi di una classe sono **`public`**, i metodi dovrebbero essere dichiarati **`private.`**

f. Se i campi di una classe sono **`public`**, anche tutti i metodi dovrebbero essere dichiarati **`public`**.

---

6 (3 punti)

Quale tra le seguenti intestazioni è corretta? Se la ritenete scorretta, correggetela.

a. **`private double pippo (num1, num2)`**

`private double pippo (int num1, double num2)`

b. **`static void pippo(char[10] c, int i)`**

`static void pippo (char[] c, int i)`

c. **`char pippo (String s, int a+b)`**

`char pippo (String s, int a)`

in chiamata `pippo ("stringa", a+b)`

d. **`public static float pippo (int x, y)`**

`public static float pippo (int x, int y)`

e. **Nessuna delle intestazioni che precedono è corretta**

---

7 (3 punti)

Considerate le seguenti classi:

```
class Veicolo { … }
class Auto extends Veicolo { … }
class SUV extends Veicolo { … }
```

Quale affermazione tra le seguenti e' vera. Spiegare in modo conciso il perche'.

1. **`Veicolo v = new Auto();`** **V** **F**

La classe `Auto` estende la classe `Veicolo` e quindi è compatibile con la dichiarazione

2. **`Veicolo v = new SUV();`** **V** **F**

La classe `SUV` estende la classe `Veicolo` e quindi è compatibile con la dichiarazione

3. **`Auto a = new SUV();`** **V** **F**

Le due classi non sono in relazione gerarchica

4. **`SUV s = new Auto();`** **V** **F**

Le due classi non sono in relazione gerarchica

---

8 (3 punti)

Considerate il seguente metodo:

```
public static void mistero (int [] a, int [] b) {
```

```
    for (int i = 0; i <a.length; i++)
        a[i] += b [b.length – 1 – i];
}
```

Calcolate i valori dell'array **a1** dopo che il codice è stato eseguito

```
int[] a1 = { 1, 3, 5, 7, 9 };
int[] a2 = { 1, 4, 9, 16, 25 };
mistero( a1, a2);
```

a1 = { 26, 19, 14, 11, 10 }