COGNOME: _____ NOME: _____ MATRICOLA: _____

Corso di Programmazione per Tecnologie Web e Multimediali
Facoltà di Scienze MM. FF. NN. — A.A. 02/03

Primo appello (17 giugno 2003)

ISTRUZIONI: Le risposte vanno scritte negli appositi riquadri, mentre i circoli ◯ possono contenere crocette. Se la risposta è troppo lunga per essere scritta nel riquadro, probabilmente è sbagliata... Ad ogni esercizio (risposta) corretta è assegnato il punteggio indicato dai numeri fra parentesi quadre, la somma dei punti vale 33. Risposte *assurde* possono dare punteggio negativo, quindi non tirate a indovinare! SCRIVETE LE RISPOSTE IN MODO CHIARO E NON AMBIGUO. Ricordatevi di apporre le vostre generalità SU OGNI FOGLIO: i compiti anonimi non verranno valutati. Consegnate anche la brutta. Tempo concesso: 2 ore.

Esercizio 1 [8] Si consideri il codice seguente:

```
class Matrice {
  public static void main (String[] args) {
    int[][] matrice = new int[7][7];
    azzera(matrice);
    cambia(matrice,5);

    for (int i = 0; i < matrice.length; i++) {
      for (int j = 0; j < matrice[i].length; j++)
        System.out.print(" " + matrice[i][j]);
      System.out.println();
    }
  }

  public static void azzera (int[][] m) {
    // ... da completare...
  }

  public static void cambia(int[][] m, int a) {
    for (int i = 0; i < a; i++) {
      m[i][i] = 1;
      m[i][m.length - i - 1] = 2;
    }
  }
}
```

(a) [3] Completate (riportandolo qui sotto) il metodo `azzera`, che deve assegnare zero a tutte le posizioni della matrice ricevuta in input.

(b) [3] Riempite la tabella sottostante con i valori della matrice prodotti in output dal programma precedente:

	0	1	2	3	4	5	6
0							
1							
2							
3							
4							
5							
6							

(c) [2] Riscrivete le ultime 5 righe del main (dalla riga vuota in poi) usando il `while` invece del `for`.

Esercizio 2 [3] Si dica quale (o quali) fra le seguenti affermazioni sono errate:

◯ Una classe astratta non può essere istanziata.

◯ Da una classe astratta non è possibile ereditare.

◯ Una classe astratta deve contenere almeno un metodo astratto.

◯ Se un metodo è astratto, la classe che lo contiene deve essere anch'essa astratta.

◯ La classe `java.awt.WindowAdapter` è astratta.

◯ La classe `java.awt.WindowAdapter` contiene metodi astratti.

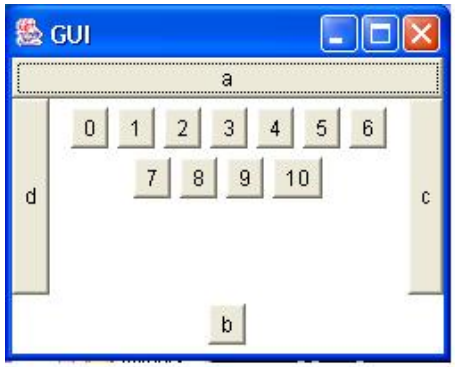
◯ I metodi in un'interfaccia sono sempre astratti.

◯ Un metodo `static` è un metodo d'istanza.

Esercizio 3 [2] I seguenti identificatori sono stati scritti rispettando le convenzioni del Java. Si dica, per ognuno di essi, se è l'identificatore di una variabile, costante, metodo, classe o interfaccia (la risposta può non essere unica...).

- `velocitaDiArrivo`
- `FunzioneDiAzzeramento`
- `VELOCITA_DI_PARTENZA`
- `m`
- `C`

Esercizio 4 [6] Si vuole creare la seguente interfaccia grafica in Java, usando l'AWT:



Per farlo, si deve completare opportunamente il codice seguente, dove indicato dai commenti:

```
import java.awt.*;
class GUI extends Frame{
  public static void main(String[] args) {
    new GUI();
  }
  public GUI(){
    super("GUI");
    add(new Button("a"),
          BorderLayout.NORTH);
             //Completa 1
    ps.add(new Button("b"));
    add(ps,BorderLayout.SOUTH);
    add(new Button("c"),
          BorderLayout.EAST);
             //Completa 2
    Panel pc = new Panel();
    add(pc, BorderLayout.CENTER);
             //Completa 3
      pc.add(new Button(""+i));
    setSize(250,200);
    setVisible(true);
  }
}
```

(completate il codice riportando qui sotto le istruzioni)

(a) [2] `// Completa 1:`

(b) [1] `// Completa 2:`

(c) [3] `// Completa 3:`

Esercizio 5 [5] Si consideri il codice seguente:

```
class Prova {
  public static void main(String[] args) {
    A x, y;
    x = new A();
    y = new C();
    x.m();
    ((C)y).m(1);
  }
}

class A {
  public void m() {System.out.println(1);}
}

class B extends A {
  public void m() {System.out.println(2);}
}

class C extends B{
  public void m(int x) {System.out.println(3);}
}
```

(a) [3] Qual è l'output prodotto?

(b) [2] Cosa succede se si elimina il cast?

Esercizio 6 [9] Si consideri il codice seguente:

```
import java.awt.*;
import java.awt.event.*;

public class BottoneGUI extends Frame {

  public static void main(String[] args) {
    BottoneGUI f = new BottoneGUI();
  }

  public BottoneGUI() {
    super("GUI con pulsante");
    Panel p = new Panel();
    Button b = new Button("Premimi di nuovo");
    p.add(b);
    this.add(p);
    Ascoltatore a = new Ascoltatore();
    b.addActionListener(a);
    pack();
    setVisible(true);
  }
}

class Ascoltatore implements ActionListener {
  boolean flag = true;
  public void actionPerformed(ActionEvent e) {
    b.setLabel(flag ? "Premimi ancora!" :
                      "Premimi di nuovo!");
    flag = !flag;
  }
}
```

(a) [3] Questo codice non compila (la compilazione restituisce un errore). Perché?

(b) [3] Come correggere questo codice facendo uso di classi interne? (servono due modifiche: descrivetele qui di seguito indicatele direttamente sul codice usando delle frecce. L'importante è che lo facciate *in modo chiaro e non ambiguo*)

(c) [3] Come correggere questo codice senza far uso di classi interne? (partendo dal programma iniziale, servono tre modifiche: descrivetele qui di seguito o indicatele direttamente sul codice usando delle frecce. L'importante è che lo facciate *in modo chiaro e non ambiguo*)