Esercizi di riepilogo
(Fondamenti di Informatica 1 – Walter Didimo)

Selezione di prove di esame al calcolatore

Esercizio 7 (*esame del 08/01/2009*) Un oggetto della classe **TabellaDiNumeri** rappresenta una tabella (cioè una matrice) di numeri reali (i numeri possono anche essere negativi). La classe ha il seguente scheletro:

```
class TabellaDiNumeri{

      /*Costruttore: crea un oggetto TabellaDiNumeri che rappresenta la tabella
      tab passata come parametro*/
      public TabellaDiNumeri (double[][] tab){…}

      /*Restituisce il numero della tabella il cui valore è più vicino a zero*/
      public double vicinoAZero (){…}

      /*Restituisce una descrizione completa della tabella sotto forma di
      oggetto String*/
      public String toString (){…}


}
```

Si chiede di:

1) Implementare la classe **TabellaDiNumeri**;
2) Implementare una classe di test per la classe **TabellaDiNumeri**, chiamata **VerificaTabellaDiNumeri**, il cui metodo **main** svolge nell'ordine le seguenti azioni:
    a. Fa inserire all'utente una tabella di numeri reali a sua scelta (l'utente sceglie sia le dimensioni della matrice sia i numeri in essa contenuti).
    b. Visualizza all'utente una descrizione completa della tabella inserita.
    c. Visualizza all'utente il numero della tabella il cui valore è più vicino a zero.

Esercizio 8 (*esame del 24/07/2009*) Un oggetto della classe **Famiglia** rappresenta un nucleo familiare. La classe **Famiglia** ha le seguenti variabili di istanza:

- **String cognome:** che memorizza il cognome della famiglia.
- **String[] nomeComponente:** ogni elemento dell'array nomeComponente memorizza il nome di un componente della famiglia.
- **int[] età:** l'elemento i-esimo dell'array età memorizza l'età del componente i-esimo, cioè età[i] memorizza l'età di nomeComponente[i].

Inoltre la classe **Famiglia** ha i seguenti costruttori e metodi:

- Un costruttore che permette di creare un oggetto **Famiglia**, ricevendo i seguenti parametri: il cognome della famiglia, i nomi dei suoi componenti (sotto forma di array di stringhe) e le rispettive età (sotto forma di array di interi).

- Un metodo di nome **piùGrande** che restituisce, come unica stringa, cognome e nome del componente della famiglia che ha l'età più grande (se ci sono più di due componenti di età massima, ne restituisce uno arbitrariamente).
- Un metodo di nome **minori** che visualizza i nomi di tutti i componenti della famiglia che sono minorenni (cioè che hanno meno di 18 anni).

Svolgere i seguenti punti:

1. Scrivere la classe **Famiglia** in base alle specifiche suddette.
2. Scrivere la classe **ProvaFamiglia**, per il test della classe **Famiglia**. Il metodo **main** della classe di test svolge i seguenti punti:
   - Fa inserire all'utente i dati completi di una famiglia (il numero di componenti è scelto dall'utente).
   - Visualizza all'utente cognome e nome del componente con età maggiore.
   - Visualizza all'utente il nome di tutti i componenti con età minore di 18 anni.

Esercizio 9 (*esame del 11/01/2008*) Un oggetto della classe **MatriceDiInteri** rappresenta una matrice di dimensioni qualunque, contenente come elementi tutti numeri interi.
La classe deve avere:

- Un costruttore che consente di creare un oggetto **MatriceDiInteri,** prendendo come parametro un array di array di **int** che specifica come è fatta la matrice.
- Un metodo di istanza, di nome **analizza**, che non ha parametri in ingresso e che restituisce sotto forma di oggetto String una descrizione comprendente: (1) il numero di numeri pari nella matrice; (2) il numero di numeri dispari nella matrice; (3) il numero di numeri divisibili per 3 nella matrice.

Scrivere la classe **MatriceDiInteri**. Scrivere inoltre una classe di test, di nome **ProvaMatriceDiInteri**, che

- Fa inserire all'utente una matrice di interi con dimensioni ed elementi a sua scelta;
- Crea un oggetto **MatriceDiInteri**, usando i dati inseriti dall'utente;
- Visualizza la descrizione restituita dal metodo **analizza**, invocato sull'oggetto creato al passo precedente.

Esercizio 10 (*esame del 29/06/2007*) Un oggetto della classe `Frase` modella una frase composta da parole senza spazi. La sequenza delle parole che formano una frase è memorizzata tramite un array di oggetti String. Lo scheletro della classe è come segue.

```
class Frase{
  private String[] frase; // memorizza le parole che formano la frase

  /* Costruttore: crea un oggetto Frase composto dalla sequenza di parole
  specificate nell'array f */
  public Frase (String[] f){…}

  /* Restituisce un array contenente le sole parole della frase che NON
  includono la sottostringa ss */
  public String[] eliminaContenenti (String ss){…}
}
```

1. Scrivere la classe `Frase`
2. Scrivere una classe `ProvaFrase` che
    a) Acquisisce dall'utente una sequenza di parole (si assuma che esse non contengano spazi), decisa dall'utente stesso.
    b) Crea un oggetto della classe `Frase` che rappresenta la sequenza di parole inserita.
    c) Chiede all'utente di specificare una ulteriore stringa ss.
    d) Visualizza all'utente tutte e sole le parole della frase inserita che NON contengono ss come sottostringa.

Selezione di prove di esame scritte

Prova del 08/01/2009

Esercizio 1. Scrivere un metodo di classe che prende in ingresso (come parametri formali) un array **a** di oggetti **String** e un ulteriore oggetto **String s,** e che visualizza sullo standard output (cioè tramite **System.out**) tutte le stringhe dell'array **a** che iniziano con la sottostringa **s**.

Esercizio 2. Rispondi in modo sintetico ma chiaro alle seguenti domande.

- Quali sono le parti principali che compongono la struttura di una classe in Java?
- Con riferimento all'assegnazione di valori a variabili di tipo primitivo in Java, in che caso si rende necessario un cast esplicito? (mostra anche un esempio).

Prova del 24/07/2009

Esercizio 1. Scrivi un metodo di classe che prende in ingresso, come parametro formale, un intero k e che restituisce in uscita una matrice quadrata di dimensione k, avente tutti 1 sulla diagonale principale e tutti 0 nelle altre celle.

Esercizio 2. Svolgi i seguenti punti.

- Scrivi un frammento di codice Java che, dato un oggetto String *s*, stampa (sullo standard output) il numero di vocali di *s* (devi assumere che *s* sia già stato definito ed inizializzato).
- Spiega la differenza tra i tipi primitivi byte, short, int e long in Java.

Prova del 11/01/2008

Esercizio 1. Scrivere un metodo di classe di nome **verificaPari**, che prende in ingresso una matrice **mat** di interi e che restituisce una nuova matrice **mat1** delle stesse dimensioni di mat e tale che, per ogni possibile posizione (i,j) si ha che mat1[i,j] vale true se mat[i,j] è pari, e false se mat[i,j] è dispari.

Esercizio 2. Rispondi sinteticamente alle seguenti domande.

1. Quale è la differenza tra classe ed oggetto? Che rapporto c'è tra i due concetti?

2. In Java, è sempre possibile sostituire un codice iterativo che usa il while con un codice iterativo che usa il for, senza alterare il risultato? (Motiva brevemente la risposta).

Prova del 29/06/2007

Esercizio 1 (**6 punti**) Scrivere un metodo di classe che prende in ingresso una matrice di oggetti String (non necessariamente quadrata) e che visualizza tutte le stringhe della matrice che iniziano e finiscono con una 'a' (minuscola o maiuscola).

Esercizio 2 (4 punti) Rispondere vero o falso a ciascuna delle seguenti affermazioni, motivando brevemente la risposta.

1) Si può usare la parola chiave **this** nel codice di un metodo di classe **[V] [F]**

Motivazione:

2) Se x è una variabile di classe, definite nell'ambito di una classe A, allora ogni istanza di A avrà una sua copia della variabile x **[V] [F]**

Motivazione:

3) Se i è una variabile intera con un certo valore assegnato prima dell'if, il seguente frammento di codice visualizzerà sempre "sarò promosso" **[V] [F]**

```
Definisce i ed assegna ad i un valore
if (i%10 < 10)
  System.out.println ("sarò promosso");
```

Motivazione: