Prova d'Esame del 13/07/2011

Esercizio 1. Si deve modellare, utilizzando il linguaggio di programmazione Java, un'agenzia di viaggi. In particolare, ciò che interessa rappresentare è l'offerta dell'agenzia, cioè l'insieme di vacanze che l'agenzia propone ai clienti. Per semplicità, supponiamo che una vacanza sia caratterizzata semplicemente dalla destinazione di viaggio, dall'importo in Euro del costo dell'intero pacchetto vacanza (viaggio + soggiorno) e dalla tipologia (una tra le seguenti: mare, monti, arte, studio, svago).
Le vacanze proposte dall'agenzia sono contenute in due diversi cataloghi: uno per le vacanze economiche, ed un altro per le vacanze lusso. Una vacanza viene considerata economica se il suo costo è minore o uguale ad una certa soglia (inizialmente posta uguale a 1000 Euro), mentre è considerata di lusso in caso contrario.
Si implementino la classe Vacanza con i metodi che si ritiene opportuno definire, e la classe AgenziaViaggi con almeno i seguenti metodi (se necessario, si potranno ovviamente definire altri metodi ausiliari):

- **costruttore**: ricevuto in input un array di vacanze, provvede a riempire i due cataloghi dell'agenzia tenendo conto della soglia di costo per le vacanze economiche.
- **stampaCataloghi**: stampa in output le vacanze incluse nei due cataloghi.
- **getVacanzeAccessibili**: ricevuto in input un importo in euro che rappresenta il budget a disposizione, restituisce l'insieme di tutte le vacanze che è possibile acquistare con tale budget.
- ***I metodi descritti di seguito sono obbligatori per gli studenti di Informatica. Gli studenti di Informatica 2 possono considerarli opzionali; agli studenti di Informatica 2 che dovessero decidere di svolgerli, e lo facessero correttamente, verrà riconosciuto un BONUS.***
  - **ordinaCataloghi**: ordina i due cataloghi per costo crescente.
  - **scegliVacanza**: ricevuto in input un importo in euro che rappresenta il budget a disposizione, seleziona (e stampa) a caso una vacanza tra quelle con costo minore o uguale all'importo ricevuto in input.

SOLUZIONE

```
public enum TipoVacanza {
    mare, monti, arte, studio, svago
}
public class Vacanza {

    private String destinazione;
    private int costo;
    private TipoVacanza tipologia;

    public String getDestinazione() {
        return destinazione;
    }
    public void setDestinazione(String destinazione) {
        this.destinazione = destinazione;
    }

    public int getCosto() {
        return costo;
    }
    public void setCosto(int costo) {
        this.costo = costo;
    }

    public TipoVacanza getTipologia() {
        return tipologia;
    }
    public void setTipologia(TipoVacanza tipologia) {
        this.tipologia = tipologia;
    }

    public String toString() {
        return getDestinazione() + ", " + getCosto() + " euro";
    }
}

public class AgenziaViaggi {

    private Vacanza[] vacanzeEconomiche;
    private Vacanza[] vacanzeLusso;
    private int soglia = 1000;

    public int getSoglia() {
        return soglia;
    }

    public void setSoglia(int soglia) {
        this.soglia = soglia;
    }

    public Vacanza[] getVacanzeEconomiche() {
        return vacanzeEconomiche;
    }
```


CAMPUS DI ARCAVACATA - Ponte Pietro Bucci Cubo 30B - 87036 Arcavacata di Rende (CS) tel. (+39) 0984 496400 fax (+39) 0984 496410
http://sv.mat.unical.it

Prova d'Esame del 13/07/2011

```
public void setVacanzeEconomiche(Vacanza[] vacanzeEconomiche) {
    this.vacanzeEconomiche = vacanzeEconomiche;
}

public Vacanza[] getVacanzeLusso() {
    return vacanzeLusso;
}

public void setVacanzeLusso(Vacanza[] vacanzeLusso) {
    this.vacanzeLusso = vacanzeLusso;
}

public AgenziaViaggi(Vacanza[] vacanze) {
    int numeroVacanzeEconomiche = 0;
    for(int i = 0; i < vacanze.length; i++) {
        if(vacanze[i].getCosto() <= this.soglia)
            numeroVacanzeEconomiche++;
    }

    vacanzeEconomiche = new Vacanza[numeroVacanzeEconomiche];
    vacanzeLusso = new Vacanza[vacanze.length - numeroVacanzeEconomiche];

    int j = 0;
    int k = 0;
    for(int i = 0; i < vacanze.length; i++) {
        if(vacanze[i].getCosto() <= this.soglia) {
            vacanzeEconomiche[j] = vacanze[i];
            j++;
        }
        else {
            vacanzeLusso[k] = vacanze[i];
            k++;
        }
    }
}

public void stampaCataloghi() {
    System.out.println("Vacanze economiche:");
    for(int i = 0; i < vacanzeEconomiche.length; i++)
        System.out.println("- " + vacanzeEconomiche[i]);

    System.out.println("\nVacanze lusso:");
    for(int i = 0; i < vacanzeLusso.length; i++)
        System.out.println("- " + vacanzeLusso[i]);
}

public Vacanza[] getVacanzeAccessibili(int budget) {
    int n = 0;
    for(int i = 0; i < vacanzeEconomiche.length; i++) {
        if(vacanzeEconomiche[i].getCosto() <= budget)
            n++;
    }
    for(int i = 0; i < vacanzeLusso.length; i++) {
        if(vacanzeLusso[i].getCosto() <= budget)
            n++;
    }

    Vacanza[] ris = new Vacanza[n];
    n = 0;
    for(int i = 0; i < vacanzeEconomiche.length; i++) {
        if(vacanzeEconomiche[i].getCosto() <= budget) {
            ris[n] = vacanzeEconomiche[i];
            n++;
        }
    }
    for(int i = 0; i < vacanzeLusso.length; i++) {
        if(vacanzeLusso[i].getCosto() <= budget) {
            ris[n] = vacanzeLusso[i];
            n++;
        }
    }

    return ris;
}

public void ordinaCataloghi() {
    ordinaCataloghiAux(vacanzeEconomiche, 0, vacanzeEconomiche.length - 1);
    ordinaCataloghiAux(vacanzeLusso, 0, vacanzeLusso.length - 1);
}
private void ordinaCataloghiAux(Vacanza[] v, int from, int to) {
    if(from < to) {
```


CAMPUS DI ARCAVACATA - Ponte Pietro Bucci Cubo 30B - 87036 Arcavacata di Rende (CS) tel. (+39) 0984 496400 fax (+39) 0984 496410
http://sv.mat.unical.it

Prova d'Esame del 13/07/2011

```
            int mid = (to + from) / 2;
            ordinaCataloghiAux(v, from, mid);
            ordinaCataloghiAux(v, mid + 1, to);
        }

        ordinaCataloghiMerge(v, from, to + 1);
    }
    private void ordinaCataloghiMerge(Vacanza[] v, int from, int to) {
        Vacanza[] aux = new Vacanza[v.length];
        int mid = (to + from) / 2;

        int i = from;
        int j = mid;
        int k = from;
        while(i < mid && j < to) {
            if(v[i].getCosto() < v[j].getCosto())
                aux[k++] = v[i++];
            else
                aux[k++] = v[j++];
        }
        while(i < mid)
            aux[k++] = v[i++];
        while(j < to)
            aux[k++] = v[j++];

        for(k = from; k < to; k++)
            v[k] = aux[k];
    }

    public Vacanza scegliVacanza(int budget) {
        Vacanza[] possibili = getVacanzeAccessibili(budget);

        Random random = new Random();
        int n = random.nextInt(possibili.length);

        return possibili[n];
    }
}
```

Esercizio 2. Si consideri il metodo riportato di seguito.

```
int met(int v [], int x) {
  if (x == 0)
    return x;
  int c = 0;
  for(int i = 0; i < v.length; i++)
    if(v[i] == x) {
      c++;
      v[i]=0;
    }
  return ( c + met(v, c) );
}
```

Si descriva sinteticamente la funzione svolta dal metodo e, in particolare, si mostri l'esecuzione e cosa viene restituito nel caso in cui viene invocato con *v = {2 3 1 3 4 2 4 9 2}* e *x=4*.

SOLUZIONE

Il metodo è RICORSIVO. Ricevuti come parametri un array di numeri naturali ed un numero naturale X, restituisce un altro numero naturale N calcolato come segue: cerca tutte le occorrenze dell'intero X, e nelle relative posizioni sostituisce X con 0; se indichiamo il numero di queste occorrenze con C, ripete poi il procedimento appena illustrato cercando le occorrenze di C... e così via. Il processo si ferma quando capita che il numero di cui ricercare le occorrenze è 0. Il numero naturale N da restituire è dato dalla somma di tutte le occorrenze dei numeri cercati.
I passi che il metodo esegue nel caso indicato sono i seguenti:
2 3 1 3 4 2 4 9 2 ← il 4 è presente 2 volte; tutte le occorrenze di 4 passeranno a 0; il nuovo numero da cercare è 2.
2 3 1 3 0 2 0 9 2 ← il 2 è presente 3 volte; tutte le occorrenze di 2 passeranno a 0; il nuovo numero da cercare è 3.
0 3 1 3 0 0 0 9 0 ← il 3 è presente 2 volte; tutte le occorrenze di 3 passeranno a 0; il nuovo numero da cercare è 2.
0 0 1 0 0 0 0 9 0 ← il 2 è presente 0 volte; il processo termina restituendo 7 (cioè 2+3+2).

Esercizio 3. Si ha un insieme S di punti su un piano, di cardinalità M. Ciascuno dei punti di S è identificato con un numero intero nell'intervallo [0, M-1], o, equivalentemente, nell'intervallo [o,M[. Si consideri ora una matrice D di interi, di dimensione M x M, che rappresenta le distanze tra i punti dell'insieme, presi per coppie. Pertanto, in particolare, il generico elemento D[i][j] della matrice rappresenta la distanza d(i, j) tra i punti i e j sul piano.
Si scriva in Java un metodo

CAMPUS DI ARCAVACATA - Ponte Pietro Bucci Cubo 30B - 87036 Arcavacata di Rende (CS) tel. (+39) 0984 496400 fax (+39) 0984 496410
http://sv.mat.unical.it

Prova d'Esame del 13/07/2011

public static boolean *verificaTriangolare*(int M[][])

che riceva in input la matrice D e restituisca *true* se la matrice rispetta la disuguaglianza triangolare, e *false* altrimenti. Si ricorda che la disuguaglianza triangolare è rispettata in S se, comunque si scelgano 3 punti *i, j* e *k*, si ha sempre che la distanza diretta da *i* a *k* è minore o uguale a quella indiretta passando per il punto intermedio *j*; formalmente:

$$d(i, k) \leq d(i, j) + d(j, k).$$

NOTA BENE: il metodo deve restituire *true* solo se la disuguaglianza è rispettata comunque si scelgano 3 punti in S.

SOLUZIONE

```
import java.util.*;

public class DisuguaglianzeTriangolari {

	public static Scanner input = new Scanner(System.in);

	public static void leggiMatrice(int m[][]) {
		System.out.println("Inserire gli elementi della matrice - " + m.length
				+ " righe, ");

		for (int i = 0; i < m.length; i++) {
			System.out.println(m[i].length + " elementi per la riga " + i);
			for (int j = 0; j < m[i].length; j++)
				m[i][j] = input.nextInt();
			System.out.println();
		}
	}

	public static void scriviMatrice(int m[][]) {
		for (int i = 0; i < m.length; i++) {
			for (int j = 0; j < m[i].length; j++)
				System.out.print(m[i][j] + " ");
			System.out.println();
		}
	}

	public static boolean verificaTriangolare(int m[][]) {
		for (int i=0; i<m.length; i++)
			for (int j=0; j<m.length; j++)
				for (int k=0; k<m.length; k++) {
					if ( i!=j && i!=k && j!=k )
						if (m[i][k] > m[i][j]+m[j][k])
							return false;
				}
		return true;
	}

	public static void main(String[] args) {
		System.out.println("Inserire le dimensioni della matrice");
		System.out.print("Numero di righe   : ");
		int nr = input.nextInt();
		System.out.print("Numero di colonne : ");
		int nc = input.nextInt();

		int m[][] = new int [nr][nc];
		leggiMatrice(m);
		scriviMatrice(m);

		System.out.println("\n\n");

		if(verificaTriangolare(m))
			System.out.println(" OK! ");
		else
			System.out.println(" NO.............! ");
	}
}
```


CAMPUS DI ARCAVACATA - Ponte Pietro Bucci Cubo 30B - 87036 Arcavacata di Rende (CS) tel. (+39) 0984 496400 fax (+39) 0984 496410
http://sv.mat.unical.it