

INFORMATICA GRAFICA - FONDAMENTI DI INFORMATICA
Ing. Civile - Edile/Architettura – Dott. Penzo

SOLUZIONE PROVA SCRITTA DEL 16/09/2002

Esercizio 4)

```
import fiji.io.SimpleReader;
class Programma{
public static void main(String[] args){

    SimpleReader in;

    Pizzeria P;
    Menu m;
    Pizza p;

    String nomePizza, ingrediente, scelta, oggi;
    String[] ingredienti;
    String nomePizzeria, indirizzo, telefono, giornoChiusura;
    int prezzo,i,j;
    Pizza[] pizzeSenza;
    boolean fine, terminato;

    in = new SimpleReader();

    // fase di inizializzazione della Pizzeria, parte non richiesta dal testo
    // ma necessaria ai fini del test
    System.out.println("Inserisci i dati della Pizzeria: ");

    System.out.print("Nome: ");
    nomePizzeria = in.readString();

    System.out.print("Indirizzo: ");
    indirizzo = in.readLine();

    System.out.print("Telefono: ");
    telefono = in.readString();

    System.out.print("Giorno di chiusura: ");
    giornoChiusura = in.readString();

    ingredienti = new String[Pizza.MAX_INGREDIENTI];

    m = new Menu();

    // inizializzazione del menu
    i = 0;
    fine = false;
    while(i < Menu.MAX_PIZZE && !fine){
        System.out.print("Nome della pizza "+(i+1)+" ": ");
        nomePizza = in.readString();

        terminato = false;
        j = 0;
        while(j < Pizza.MAX_INGREDIENTI && !terminato){
            System.out.print("Ingrediente "+(j+1)+" ": ");
            ingredienti[j++] = in.readString();

            System.out.print("Altro ingrediente? (si/no): ");
            scelta = in.readString();
```

```

        terminato = scelta.equals("no");
    }
    System.out.print("Prezzo della pizza: ");
    prezzo = in.readInt();

    p = new Pizza(nomePizza, ingredienti, prezzo);
    m.inserisciNuovaPizza(p);

    System.out.print("Altra pizza? (si/no): ");
    scelta = in.readString();
    fine = scelta.equals("no");

    i++;
}

P = new Pizzeria(nomePizzeria, indirizzo, telefono, giornoChiusura, m);

// punto 1)
// se c'e` ancora spazio per una pizza
if (P.numPizzeInMenu() < P.maxNumPizzeInMenu()){
    System.out.print("Inserisci il nome di una nuova pizza: ");
    nomePizza = in.readString();

    terminato = false;
    j = 0;
    while(j < Pizza.MAX_INGREDIENTI && !terminato){
        System.out.print("Ingrediente "+(j+1)+" : ");
        ingredienti[j++] = in.readString();

        System.out.print("Altro ingrediente? (si/no): ");
        scelta = in.readString();
        terminato = scelta.equals("no");
    }
    System.out.print("Prezzo della pizza: ");
    prezzo = in.readInt();

    p = new Pizza(nomePizza, ingredienti, prezzo);
    P.aggiungiPizza(p);
}
else
    System.out.println("Non c'e` piu` spazio per inserire nuove
pizze!");

// punto 2)
System.out.print("Che giorno e` oggi?: ");
oggi = in.readString();
giornoChiusura = P.getGiornoDiChiusura();

if (giornoChiusura.equals(oggi))
    System.out.println("Oggi la Pizzeria "+ P.getNome() +" e` chiusa!");
else {
    // punto 3)
    System.out.print("Inserire l'ingrediente non desiderato: ");
    ingrediente = in.readString();

    // punto 4)
    pizzeSenza = P.pizzeSenza(ingrediente);

    System.out.println("Le pizze senza "+ ingrediente +" sono:");
    i = 0;
    while(i < P.maxNumPizzeInMenu() && pizzeSenza[i] != null){
        System.out.println(pizzeSenza[i]);
        i++;
    }
}

```

```

        }
    }
}

```

Esercizio 1)

```

class Pizza{
    private String nome;
    public static final int MAX_INGREDIENTI=10;
    private String[] ingredienti;
    private int prezzo;

    /**
     * Costruttore con parametri
     */
    public Pizza(String nome, String[] componenti, int prezzo){
        int i=0;

        this.nome = nome;
        this.ingredienti = new String[MAX_INGREDIENTI];
        for(i=0;i<MAX_INGREDIENTI && componenti[i] != null;i++)
            this.ingredienti[i] = componenti[i];
        this.prezzo = prezzo;
    }

    /**
     * Gets the nome
     */
    public String getNome(){
        return(nome);
    }

    /**
     * Gets the prezzo
     */
    public int getPrezzo(){
        return(prezzo);
    }

    // metodo di utilità
    /**
     * trovatoIngrediente
     */
    public boolean trovatoIngrediente(String ingrediente){
        boolean trovato = false;
        int i = 0;

        while(i < MAX_INGREDIENTI && ingredienti[i] != null && !trovato){
            if (ingredienti[i].equals(ingrediente))
                trovato = true;
            i++;
        }
        return(trovato);
    }
}

```

```

/**
 * toString
 */
public String toString(){
    return("Pizza "+ nome +" prezzo: "+prezzo);
}

}

```

Esercizio 2)

```

class Menu{

    private int numPizze;
    public static final int MAX_PIZZE=100;
    private Pizza[]  pizze;

    /**
     * Costruttore
     */
    public Menu(){
        this.pizze = new Pizza[MAX_PIZZE];
        this.numPizze = 0;
    }

    /**
     * inserisciNuovaPizza
     */
    public void inserisciNuovaPizza(Pizza pizza){
        if (numPizze < MAX_PIZZE)
            pizze[numPizze++] = pizza;
        else
            System.out.println("Non c'è più spazio per le pizze!");
    }

    /**
     * trovaPizzeSenzaIngrediente
     */
    public Pizza[] trovaPizzeSenzaIngrediente(String ingrediente){
        Pizza[] pizzeSenza = new Pizza[MAX_PIZZE];
        int numPizzeSenza = 0, i;

        for (i=0; i<numPizze; i++)
            if (!(pizze[i].trovatoIngrediente(ingrediente)))
                pizzeSenza[numPizzeSenza++] = pizze[i];
        return(pizzeSenza);
    }

    // metodo di utilità
    /**
     * Gets the numPizze
     */
    public int getNumPizze(){
        return(numPizze);
    }
}

```

Esercizio 3)

```
class Pizzeria{
    private String nome;
    private String indirizzo;
    private String telefono;
    private String giornoDiChiusura;
    private Menu menu;

    /**
     * Costruttore con parametri
     */
    public Pizzeria(String nome, String indirizzo, String telefono, String
giornoDiChiusura, Menu menu){
        this.nome = nome;
        this.indirizzo = indirizzo;
        this.telefono = telefono;
        this.giornoDiChiusura = giornoDiChiusura;
        this.menu = menu;
    }

    /**
     * aggiungiPizza
     */
    public void aggiungiPizza(Pizza pizza){
        menu.inserisciNuovaPizza(pizza);
    }

    /**
     * Sets the giornoDiChiusura
     */
    public void setGiornoDiChiusura(String giornoDiChiusura){
        this.giornoDiChiusura = giornoDiChiusura;
    }

    /**
     * Gets the giornoDiChiusura
     */
    public String getGiornoDiChiusura(){
        return(giornoDiChiusura);
    }

    // metodo di utilità
    /**
     * numPizzeInMenu
     */
    public int numPizzeInMenu(){
        return(menu.getNumPizze());
    }

    // metodo di utilità
    /**
     * maxNumPizzeInMenu
     */
    public int maxNumPizzeInMenu(){
        return(menu.MAX_PIZZE);
    }
}
```

```
// metodo di utilità
/**
 * Gets the nome
 */
public String getNome(){
    return(nome);
}
```

```
// metodo di utilità
/**
 * pizzeSenza
 */
public Pizza[] pizzeSenza(String ingrediente){
    return(menu.trovaPizzeSenzaIngrediente(ingrediente));
}
}
```