ΛΥΣΗ ΓΙΑ ΤΟ SHOPPING CART

TESTSHOPPING (ΠΕΡΙΕΧΕΙ ΤΗΝ MAIN)

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package testshopping;

import java.util.Scanner;

import javax.swing.JOptionPane;

/**
 *
 * @author LIBRARY
 */
public class TestShopping {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        ShoppingCart myCart = new ShoppingCart();
        String itemName;
        double itemPrice;
        int itemQuantity;
        int e = 0;
        Scanner sc = new Scanner(System.in);
        while (true) {
            System.out.println("1. Agora  \n2. Print cart contents  \n3. Print cart total cost 
\n4. exit");
            e = sc.nextInt();

            switch (e) {
                case 1:

                    System.out.println("Give the item Name");
                    itemName = sc.next();
                    System.out.println("Give the item price");
                    itemPrice = sc.nextDouble();
                    System.out.println("Give the item quantity");
                    itemQuantity = sc.nextInt();
                    myCart.addToCart(itemName, itemPrice, itemQuantity);
                    break;
                case 2:
                    System.out.println(myCart.toString());
                    break;
                case 3:
                    System.out.println("Total Cost so far from shopping" +
myCart.getCost());
                    break;
                case 4:
                    System.out.println(1);
                    break;
                default:
                    System.out.println("Wrong choice");
                    break;
            }
        }
    }
}
```

ITEM

```
package testshopping;

// ************************************************************
// Item.java
//
// Represents an item in a shopping cart.
// ************************************************************
import java.text.NumberFormat;

public class Item {

    private String name;
    private double price;
    private int quantity;
// -------------------------------------------------------
// Create a new item with the given attributes.
// -------------------------------------------------------
    public Item(String itemName, double itemPrice, int numPurchased) {
        name = itemName;
        price = itemPrice;
        quantity = numPurchased;
    }
// -------------------------------------------------------
// Return a string with the information about the item
// -------------------------------------------------------
    public String toString() {
        NumberFormat fmt = NumberFormat.getCurrencyInstance();
        return (name + "\t" + fmt.format(price) + "\t" + quantity + "\t"
                + fmt.format(price * quantity));
    }
// -------------------------------------------------
// Returns the unit price of the item
// -----------------------------------------------
    public double getPrice() {
        return price;
    }
// -----------------------------------------------
// Returns the name of the item
// -----------------------------------------------
    public String getName() {
        return name;
    }
// -----------------------------------------------
// Returns the quantity of the item
// -----------------------------------------------
    public int getQuantity() {
        return quantity;
    }
}
```

SHOPPINGCART

```
package testshopping;

// ***************************************************************
// ShoppingCart.java
//
// Represents a shopping cart as an array of items
// ***************************************************************
import java.text.NumberFormat;

public class ShoppingCart {

    private Item[] cart;
    private int itemCount; // total number of items in the cart
    private double totalPrice; // total price of items in the cart
    private int capacity; // current cart capacity
// -----------------------------------------------------
// Creates an empty shopping cart with a capacity of 5
// items.
// -----------------------------------------------------

    public ShoppingCart() {
        capacity = 5;
        itemCount = 0;
        totalPrice = 0.0;
        cart = new Item[5];
    }
// -----------------------------------------------------
// Adds an item to the shopping cart.
// -----------------------------------------------------
```

```
    public void addToCart(String itemName, double price, int quantity) {
        if (itemCount == cart.length) {
            increaseSize();
        }
        cart[itemCount] = new Item(itemName, price, quantity);
        totalPrice += quantity * price;
        itemCount++;
    }
    public double getCost(){
        return totalPrice;
    }
// ----------------------------------------------------
// Returns the contents of the cart together with
// summary information.
// ----------------------------------------------------

    public String toString() {
        NumberFormat fmt = NumberFormat.getCurrencyInstance();
        String contents = "\nShopping Cart\n";
        contents += "\nItem\t\tUnit Price\tQuantity\tTotal\n";
        for (int i = 0; i < itemCount; i++) {
            contents += cart[i].toString() + "\n";
        }
        contents += "\nTotal Price: " + fmt.format(totalPrice);
        contents += "\n";
        return contents;
    }
// ----------------------------------------------------
// Increases the capacity of the shopping cart by 3
// ----------------------------------------------------

    private void increaseSize() {
        Item[] temp = new Item[cart.length + 3];
        for (int i = 0; i < cart.length; i++) {
            temp[i] = cart[i];
        }
        cart = temp;
    }
}
```

ΛΥΣΗ ΓΙΑ ΤΟ ACCOUNT-PERSON-BANK

```
public class Account {

        private String code;
        private double balance;
        Person owner ;

        //constructor
```

```
    public Account(){
        code="";
        owner = new Person();
        balance = 0.0;
    }

    public void setCode(String c){
    code = c;
    }

    public String getCode(){
    return code;
    }

    // setter kai getter methods gia ton katoxo toy logariasmoy
    public void setOwner(Person p){
    owner = p;
    }

    public Person getOwner() {
    return owner;
    }

    //methodos getBalance
    public double getBalance() {
    return balance;
    }

    //methodos withdraw
    //An i analipsi ginei epistrefei true alliws false
    public boolean withdraw(double money) {
        if (money <= balance )     {
            balance -= money;
            return true;
        }
    return false;
    }

    //An to poso einai thetiko ginetai i katathesi kai
    //epistrefetai true, diaforetika epistrefetai false
    public boolean deposit(double money) {
        if (money >= 0)
        {
        balance += money;
        return true;
        }
    return false;
    }
}


public class Person {
    //Ena atomo theoroyme oti borei na exei to poli 10
    //logariasmoys.

    private String name;
    private int age;
    private final int MAX_ACCOUNTS=10;
    private Account accounts[] = new Account[MAX_ACCOUNTS];

    public Person() {
        age = 0;
        name = "";
    }

    //set kai get methods gia to name
    public void setName(String s) {
    name = s;
    }

    public String getName() {
    return name;
    }
    //set kai get methods gia to age
    public void setAge(int a) {
        age = a;
    }

    public int getAge() {
    return age;
    }

    //indexed setter kai getter methods gia ton pinaka accounts
    public void setAccount(int index, Account a){
        if (index>=0 && index <MAX_ACCOUNTS)
            accounts[index] = a;
    }

    public Account getAccount(int index) {
    //an o index einai metaxy 0 kai MAX_ACCOUNTS
    //epestrepse ton account pou vrisketai sti thesi index

    if (index>=0 && index <MAX_ACCOUNTS)
        return accounts[index];
    //diaforetika epestrepse null
    else return null;
    }

    //oxi indexed setter kai getter methods gia ton pinaka accounts
    public void setAccount(Account[] a)
    {
    accounts = a;
    }

    public Account[] getAccount()
    {
    return accounts;
    }
}


public class TestBank {
    public static void main (String[] args) {

        Person pers1 = new Person();
        pers1.setName("George");
        pers1.setAge(33);

        Person pers2 = new Person();
        pers2.setName("Natalia");
        pers2.setAge(31);

        Account ac1 = new Account();
        Account ac2 = new Account();
        Account ac3 = new Account();

        //Sto logariasmo ac1 kai ac2 dikaiouxo einai o George
        //Sto logariasmo ac3 dikaiouxos einai h Natalie
        //Oi akolouthes kliseis dhmiourgoun aytes tis syndeseis
        pers1.setAccount(0, ac1);
        pers1.setAccount(1, ac2);

        ac1.setOwner(pers1);
        ac2.setOwner(pers1);

        pers2.setAccount(0, ac3);
        ac3.setOwner(pers2);

        // kiniseis logmoy George
        ac1.deposit(10000);
        ac1.withdraw(5000);
```

```
                        ac1.withdraw(6000);

                        ac2.deposit(7000);

                        ac3.deposit(1000);
                        ac3.withdraw(500);

                        //To array a tha exei toys logariasmous toy George
                        Account[] a = pers1.getAccount();
                        System.out.println("Total accounts for George:");
                        int total=0;
                        for (int i=0; i<a.length; i++)
                                    if (a[i] != null)
                                                total += a[i].getBalance();
                        System.out.println(total);

            }
}
```

ΛΥΣΗ ΓΙΑ ΤΟ ΤΡΙΓΩΝΟ

```
// Trigono.java
public class Trigono {
   private Point a;
   private Point b;
   private Point c;

   // no-argument constructor
   public Trigono()   {
      // implicit call to Object constructor occurs here
   }

   // constructor
   public Trigono( Point p1, Point p2, Point p3)   {
      // implicit call to Object constructor occurs here
      a = new Point(p1.getX(),p1.getY());
      b = new Point(p2.getX(),p2.getY());
      c = new Point(p3.getX(),p3.getY());
   }

   // ypologismos mikoys
   public double mikos( Point pStart, Point pEnd ) {
      return (Math.sqrt(Math.pow((pStart.getX()-pEnd.getX()),2) + Math.pow((pStart.getY() -
pEnd.getY()),2)));
   }
            public double perimetros() {
                        return mikos(a,b)+mikos(b,c)+mikos(c,a);
   }
    public String toString()   {
      return "Korifi 1h[" + a.getX() + ", " + a.getY() + "]"+"\n"
      +"Korifi 2h[" + b.getX() + ", " + b.getY() + "]"+"\n"
      +"Korifi 3h[" + c.getX() + ", " + c.getY() + "]"+"\n"
      +"perimetros"+perimetros();
   }

} // end class Trigono


public class TestTrigono {
    public static void main(String[] args) {
        // Δημιουργία 1ης κορυφής
                        Point p1= new Point(10,50);

                        // Δημιουργία 2ης κορυφής
                        Point p2= new Point(70,150);

                        // Δημιουργία 3ης κορυφής
                        Point p3= new Point(200,20);
                        // Δημιουργία τριγώνου
                        Trigono tr1 = new Trigono(p1,p2,p3);

                        //εκτύπωση των ιδιοτήτων
                        System.out.println(tr1.toString());
                           }     }
```