

Übungen zu Einführung in die Informatik

Freiwillige Abgabe bis zum 03. 12. 2007, 12:00h

Aufgabe 6-1

Point

In dieser Aufgabe betrachten wir die Klasse `Point`, die Sie aus der Vorlesung kennen.

- Erweitern Sie die Klasse `Point` um die Methoden `public void setX(int x)` und `public void setY(int y)`, die dem Attribut `x` bzw. `y` einen neuen Wert zuweisen.
- Erweitern Sie die Klasse `Point` um eine Methode `public String toString()`, die ein `String`-Objekt zurückgibt, das einen (zweidimensionalen) Punkt in textueller Form beschreibt. Der Punkt `(3,5)` kann beispielsweise mit `Point [x=3, y=5]` dargestellt werden. Testen Sie die Methoden `setX()` und `setY()` mit `toString()`.
- Erweitern Sie auch das UML-Klassendiagramm aus der Vorlesung um die neuen Methoden.
- Was wird jeweils ausgegeben, wenn die mit `(*)` markierten Zeilen erreicht werden? Erklären Sie die Funktionsweisen von `shallowSwap` und `deepSwap` anhand von Speicherzustandsdiagrammen.

```
public class PointSwap{

    /**
     * This method cannot swap the points p1 and p2. Why?
     */
    public static void shallowSwap(Point p1, Point p2){
        Point t = p1;
        p1 = p2;
        p2 = t;
    }

    /**
     * This method swaps the points p1 and p2. How does it work?
     */
    public static void deepSwap(Point p1, Point p2){
        Point t = new Point(p1.getX(), p1.getY());
        p1.setX(p2.getX());
        p1.setY(p2.getY());
        p2.setX(t.getX());
        p2.setY(t.getY());
    }

    public static void main(String[] args){
        Point p1 = new Point(100,100);
        Point p2 = new Point(10,10);
        System.out.println("p1: "+p1);
        System.out.println("p2: "+p2);

        System.out.println("Shallow swapping p1 and p2");
        shallowSwap(p1, p2);
        System.out.println("p1: "+p1);           // (*)
        System.out.println("p2: "+p2);           // (*)

        System.out.println("Deep swapping p1 and p2");
        deepSwap(p1, p2);
        System.out.println("p1: "+p1);           // (*)
        System.out.println("p2: "+p2);           // (*)
    }
}
```

Aufgabe 6-2

Integer

Gegeben sei das folgende Programm

```
public class IntSwap{

    // This method should exchange the values of two int variables.
    public static void swap(int a, int b){
        int t = a;
        a = b;
        b = t;
    }

    public static void main(String args[]){
        int x = 10;
        int y = 100;
        System.out.println("x= " + x + " y= " + y);
        swap(x,y);
        System.out.println("x= " + x + " y= " + y);    // (*)
    }
}
```

Was wird ausgegeben, wenn die mit (*) markierte Zeile erreicht wird? Können Sie die Methoden `swap` verbessern, so dass ein Aufruf die Werte zweier `int`-Variablen gegeneinander austauscht?

Aufgabe 6-3

Bankkonto

- a) Implementieren sie eine Klasse `BankAccount`, welche die folgenden Eigenschaften und Funktionalitäten besitzt:
- Das Konto hat eine Kontonummer vom Typ `int`. Die Kontonummer wird dem Konto bei seiner Erzeugung zugewiesen und kann mittels der Methode `public int getAccountNumber()` erfragt werden.
 - Das Konto speichert den aktuellen Kontostand, der mittels der Methode `public double getBalance()` erfragt werden kann.
 - Auf das Konto kann mittels der Methode `public void deposit(double amount)` Geld eingezahlt werden.
 - Vom Konto kann mittels der Methode `public double withdraw(double amount)` Geld abgehoben werden. Durch eine Auzahlung darf der Kontostand nicht negativ werden, d. h. es wird maximal Geld bis zu einem Kontostand von 0 ausbezahlt. Die Methode gibt die Summe zurück, die tatsächlich ausgezahlt wird.
- b) Laden Sie die Klasse `Customer` von unserer Web-Seite herunter und geben Sie den jeweiligen Speicherzustand auf Keller und Halde zu den Zeitpunkten von (1) und (3) sowie nach dem Anruf in der mit (2) markierten Zeile an.