

Übungen zu Einführung in die Informatik

Hinweis: Für alle. Freiwillige Abgabe bis zum Montag, den 07.01.08, 12.00

Aufgabe 10-1

Point3D

Laden Sie die Klasse `Point`, die Sie aus Übungsblatt 5 kennen, von unserer Web-Seite herunter und erweitern Sie sie wie folgt:

- Definieren Sie eine Subklasse `Point3D` von `Point`, die eine dritte Koordinate `int z` enthält. Implementieren Sie eine Getter- und eine Setter-Methode für `z`.
- Implementieren Sie einen Konstruktor für `Point3D`, der alle drei Werte für `x`, `y` und `z` übergeben bekommt und die Attribute entsprechend setzt. Dabei soll der Konstruktor von dem der Klasse `Point` Gebrauch machen.
- Implementieren Sie in der Klasse `Point3D` die Methode `public String toString()`, die eine angemessene textuelle Darstellung des impliziten Parameters zurückgibt.
- Implementieren Sie eine Klasse `Point3DUtil`, die folgende Methoden enthalten soll:
 - `public static void sortXAsc (Point3D[] points)`, die die Elemente des Array `points` nach dem `x`-Wert *aufsteigend* sortiert.
 - `public static void sortYDsc (Point3D[] points)`, die die Elemente des Array `points` nach dem `y`-Wert *absteigend* sortiert.
 - `public static void sortZDsc (Point3D[] points)`, die die Elemente des Array `points` nach dem `z`-Wert *absteigend* sortiert.

Verwenden Sie für die drei Methoden bitte drei verschiedene Sortieralgorithmen. Geben Sie per Kommentar an, welcher Algorithm in welcher Methode verwendet wird.

- Lassen Sie in der `Main`-Methode 20 `Point3D`-Objekte mit zufallsgenerierten `x`-, `y`- und `z`-Werten erzeugen (Hinweise: `Math.random()`) und testen Sie Ihre Methoden.

Aufgabe 10-2

Studenten

An einer Universität studieren Studenten. Jeder Student hat einen Namen und eine Fachrichtung.

- Mollieren Sie die Klassen `University` und `Student` sowie deren Beziehung in einem Klassendiagramm. Geben Sie insbesondere Multiplizitäten und Rollennamen an.
- Implementieren Sie die Klasse `Student` in Java. Implementieren Sie für alle Attribute entsprechende Getter- und Setter-Methoden. Implementieren Sie eine Methode `public String toString()`, die eine textuelle Darstellung des Studenten zurückgibt.
- Implementieren Sie die Klasse `University` in Java. Sie soll insbesondere eine Methode `public void addStudent(Student st)` enthalten, womit der Student `st` in die Studentenliste der Universität aufgenommen werden kann. Achten Sie auf Datenkonsistenz: wenn eine Universität `uni` einen Studenten `st` enthält, soll dies auch im Zustand von `st` ablesbar sein.
- Implementieren Sie in der Klasse `University` eine Methode `public void sortStudents()`, welche die Studenten nach Fachrichtung als Erstkriterium und Namen als Zweitkriterium sortiert. Beispielsweise sollte nach der Sortierung der Informatikstudent `Zhang` vor dem Mathematikstudenten `Böhm`, aber nach dem Informatikstudenten `Mayer` stehen. Sie können dabei einen Sortieralgorithmus Ihrer Wahl verwenden.
- Testen Sie Ihr Programm.

Hinweise

- Die Methode `compareTo(String s)` der Klasse `String` vergleicht den impliziten Parameter `this` mit dem Parameter `s`. Sie gibt eine negative ganze Zahl zurück, falls `this` alphabetisch vor `s` steht; eine positive ganze Zahl, falls `this` alphabetisch hinter `s` steht; und 0, falls `this` und `s` gleich sind. Siehe [http://java.sun.com/j2se/1.4.2/docs/api/java/lang/String.html#compareTo\(java.lang.String\)](http://java.sun.com/j2se/1.4.2/docs/api/java/lang/String.html#compareTo(java.lang.String))
- Es wird angenommen, dass ein Student nur an einer Universität studiert.

Wir wünschen allen Studenten dieser Vorlesung ein fröhliches Weihnachtsfest und einen guten Rutsch ins neue Jahr!