

Übungsblatt 4

Abgabe bis Donnerstag, 02.06.2011, 12:00 Uhr

Hinweis:

Programmieraufgaben immer per E-Mail (eine E-Mail pro Blatt und Gruppe) an den zuständigen Tutor schicken (Java Quellcode und eventuell benötigte Datendateien). Schriftliche Lösungen bitte ebenfalls an den zuständigen Tutor schicken oder in die Briefkästen in Geb. 051, Erdgeschoss einwerfen.

Aufgabe 4.1

Betrachten Sie den folgenden Auszug der Klasse `Student`, die eine Studentin bzw. einen Studenten repräsentiert. Der Vorname, Nachname sowie der Studiengang wird dabei in den entsprechenden Instanzvariablen `firstName`, `surname` und `subject` gespeichert.

```
public class Student {  
  
    public Student(String firstName, String surname, String subject) {  
        ...  
    }  
  
    public void setFirstName(String s) { ... }  
  
    public void setSurname(String s) { ... }  
  
    public void setSubject(String s) { ... }  
  
    public String getFirstName() { ... }  
  
    public String getSurname() { ... }  
  
    public String getSubject() { ... }  
  
    public String toString() { ... }  
  
    private String firstName;  
    private String surname;  
    private String subject;  
  
}
```

1. Identifizieren Sie Konstruktoren, Methoden, Instanzvariablen und Argumente.
2. Vervollständigen Sie den Konstruktor der Klasse `Student`.
3. Vervollständigen Sie die Methoden `setFirstName`, `setSurname` sowie `setSubject` zum Setzen und `getFirstName`, `getSurname` und `getSubject` zum Auslesen der Instanzvariablen.
4. Vervollständigen Sie die Methode `toString`, die eine Beschreibung der Studentin/des Studenten in der Form `Max Mustermann studiert Informatik` zurückgibt.
5. Schreiben Sie eine `main`-Methode, um Ihre Klasse zu testen. Erzeugen Sie dafür ein Instanz der Klasse `Student` mit Ihren Daten und geben Sie dessen Beschreibung mit der Methode `toString` auf dem Monitor aus.

Aufgabe 4.2

1. Implementieren Sie eine Klasse `MathString`, die folgende Methoden zur Erzeugung mathematischer Ausdrücke enthält:
 - `public String add(String s1, String s2) { ... }`
Diese Methode soll die als Argumente übergebenen Strings mit "+" verknüpfen und zurückgeben. (Bsp: `add("1", "2")` gibt `"1 + 2"` zurück)
 - `public String subtract(String s1, String s2) { ... }`
Diese Methode soll die als Argumente übergebenen Strings mit "-" verknüpfen und zurückgeben. (Bsp: `subtract("1", "2")` gibt `"1 - 2"` zurück)
 - `public String bracket(String s) { ... }`
Diese Methode soll den als Argument übergebenen String in Klammern setzen und zurückgeben. (Bsp: `bracket("1")` gibt `"(1)"` zurück)
 - `public String equals(String s1, String s2) { ... }`
Diese Methode soll die als Argumente übergebenen Strings mit "=" verknüpfen und zurückgeben. (Bsp: `bracket("1", "1")` gibt `"1 = 1"` zurück)
2. Schreiben Sie eine `main`-Methode, um die eben implementierte Klasse `MathString` zu testen. Erzeugen Sie dafür eine Instanz der Klasse `MathString` und benutzen Sie deren Methoden, um den String `"20 - (7 + 5) = 8"` auszugeben.