Übung 12

Objektorientierung

Aufgabe 1: Entwerfen Sie eine Klasse `Student` nach folgendem Muster (3 Punkte)

```
public class Student {

 // Hier muss etwas ergänzt werden

 private int matrikelnummer;
 private String vorname;
 private String nachname;

 // Konstruktor
 public Student(String vorname, String nachname){

  // Hier muss etwas ergänzt werden:
  // matrikelnummer aus Vorlage übernehmen und Vorlage hochzählen

  // private Variablen vorname und nachname auf die im Konstruktor
  // übergebenen Werte setzen



 }

 public String toString(){
  // Hier als Returnwert ein String-Objekt wie folgt zurückliefern:
  // Vorname: (Variable), Nachname: (Variable),
  // matrikelnummer: (Variable)


 }

 // 3 Funktionen zum Auslesen der privaten Variablen:
 public int get_matrikelnummer()   {                          }
 public String get_vorname()       {                          }
 public String get_nachname()      {                          }
}
```

- Beim Instanzieren eines **`Student`** Objektes soll im Konstruktor der Name des Studenten übergeben und in die privaten Variablen kopiert werden (d.h. Aufruf eines Kopierkontruktors notwendig!). Die Matrikelnummer soll **automatisch hochlaufend** erzeugt werden und **mit 100000 beginnen**.
- Durch drei weitere Funktionen sollen die privaten Variablen ausgelesen werden können, um sie z.B. im Formularen direkt weiterverarbeiten zu können.
- Überschreiben Sie die Methode **String toString()**, um eine textuelle Repräsentation des **`Student`**-Objektes zurückliefern zu können.

Aufgabe 2: Aufbau einer Studentenverwaltung für Vorlesungen (6 Punkte)

Programmieren Sie eine Studentenverwaltung nach folgendem Muster:

```
public class Vorlesung {

 private String titel;
 private Student[] teilnehmer;
 private int max_teilnehmer;

 public Vorlesung(String titel, int max_teilnehmer){
  // Vorlesungstitel kopieren,
  // leeres Array erzeugen,
  // private Variable max_teilnehmer setzen

 }

 public int studentenZaehlen(){
  // Gibt die Anzahl der Elemente des teilnehmer-Arrays
  // zurück, die nicht null sind.



 }

 public boolean studentHinzufuegen(Student student){
  // Wenn Anzahl Teilnehmer größer als max_teilnehmer
  // -> Warnung ausgeben, und keinen neuen Eintrag durchführen.

 }

 public int studentIndex(Student student){
  // Arrayelement für Student suchen, Position zurückgeben
```

```
  // -1 falls nicht gefunden.


 }

 public boolean studentEntfernen(int index){
  // Student an index im Array nullsetzen

 }

 public void printTeilnehmer(){
  // Alle Teilnehmer auflisten, die an der Vorlesung teilnehmen
  // dürfen (max_teilnehmer und null-Elemente beachten)


 }

}
```

- Implementieren Sie in der Klasse **`Vorlesung`** die angegebenen Methoden gemäß ihrer Beschreibung.

Hinweise zur Abgabe:

- Abgabe der Lösungen für diese Übung per E-Mail an Herrn Marc Beck <marc.beck@hs-kl.de>
  bis **Mittwoch, 29.06.2016, 23:59**.
- Bitte geben Sie als „Betreff“ an: **Abgabe GDI Übung 12 SS2016 *Ihr Name***.