

Blatt 1

Punktetabelle

1	2	Σ
5	10	15

1 Aufgabe 1 - Präzedenzen

Mit dieser geänderten Präzedenzliste sollen die darunter stehenden Ausdrücke vollständig geklammert und dann ausgewertet werden.

Präzedenz	Operator	Assoziativität
5	−, +	rechts
4	^	links
3		links
2	&&	links
1	*, /	rechts

- **true || false && false**
- **3 + 5 * 7**
- **6 / 3 * 2**
- **100 − 30 − 14**
- **true ^ true && true || false && true**

2 Aufgabe 2 - Studentenverwaltung

Der König der Insel Fani möchte eine neue Universität gründen. Diese Universität braucht ein neues Verwaltungsprogramm. In erster Instanz soll es die Studierenden der Universität verwalten können. Da für den König nur die Besten der Besten gut genug sind, hat er sich an euch gewendet, dieses Programm zu implementieren. Von dem Studierenden sollen die Matrikelnummer, das Geburtsdatum, der Name und die Studienrichtung erfasst werden können.

Der Hofsoftwaredesigner stellt dazu noch folgende Anforderungen an das Programm: Es soll Java verwendet werden. Für die Implementierung sollen nur primitive Datentypen und keine Objekte, ausgenommen Strings und Arrays, verwendet werden. Die Daten von den Studierenden sollen in Arrays gehalten werden. Dazu sollen folgende Arrays verwendet werden:

```
static String [] birthDay = new String [0]; // globally accessible data
static String [] name = new String [0];
static int [] skz = new int [0];
static int [] state = new int [0]; //0 = immatr., 1=exmatr., 2=finished
```

Da die Universität neu eröffnet, kann man davon ausgehen, dass die Matrikelnummern beginnend ab 1 fortlaufend vergeben werden, und somit kann man die Matrikelnummer als Index für die Arrays verwenden. Weiter soll das Verwaltungsprogramm Studierende immatrikulieren und exmatrikulieren, sowie die Stammdaten der Studierenden ändern können. Hierfür sind folgende Funktionen zu implementieren.

```
/**
 * This method matriculates a new student
 * @param name The name of the student
 * @param birthday The birthday of the student
 * @param skz The SKZ
 */
static void matriculate( String newName, String bDay, int newSkz )
{
    String [] tmpBday = new String[birthday.length + 1];
    ...
    for (int i = 0; i < birthday.length; i++) {
        tmpBday[i] = birthday[i];
        ...
    }
    tmpBday[birthday.length] = bDay;
    ...
    birthday = tmpBday;
    ...
}

/**
 * This method updates the personal data of a student
 * @param matriculationNumber The matriculation number
 */
static void updateAccount( int matriculationNumber, String
newName, String bDay, int newSkz ) {
    ...
}

/**
 * This method exmatriculates a student.
 * @param matriculationNumber The matriculation number
 */
static void exmatriculate( int matriculationNumber ) {
    ...
}

/**
 * This method sets a student's state as finished
 * @param matriculationNumber The matriculation number
 */
static void finish( int matriculationNumber ) {
    ...
}
```

Exmatrikulierte Studierende sollen nicht gelöscht werden, sondern markiert, so dass sie jederzeit beim immatrikulieren wieder ihre alte Matrikelnummer zugewiesen bekommen. Auch dies sollte im Programm möglich sein. Des Weiteren sollen Studierende, die ein Studium erfolgreich abgeschlossen haben markiert werden. Das Programm soll auf Anfrage hin alle Studierenden die noch am Studieren sind oder die bereits ein Studium abgeschlossen haben ausgeben können. Hierfür ist noch folgende Methode zu implementieren:

```
/**
 * This method prints a list of students.
 * @param searchState The state of the students
 */
static void printStudents( int searchState ) {
    ...
}
```