

# Programmierung für Mathematik HS12

## Übung 9

19. November 2012

### 1 Aufgabe: Rekursion

#### 1.1 Ziele

- Rekursion umsetzen können

#### 1.2 Aufgabenstellung

Schreiben Sie ein Programm mit einer Methode `fibonacci`. Diese Methode soll eine ganze Zahl  $n$  erwarten und dann das  $n$ -te Glied der Fibonacci-Folge zurückliefern. Benutzen Sie für die Implementierung keine Iteration sondern eine Rekursion.

*Hinweis:* Die Fibonacci-Folge ist wie folgt definiert:

$$x_0 = 0$$

$$x_1 = 1$$

$$x_n = x_{n-1} + x_{n-2}$$

## 2 Aufgabe: Determinante

### 2.1 Ziele

- Rekursion vertiefen

### 2.2 Aufgabenstellung

Erweitern Sie die Klasse `Matrix` aus der Übung 7 mit der Methode `det`. Diese Methode soll, falls die Matrix quadratisch ist, die Determinante der Matrix zurückliefern. Andernfalls soll die Methode 0 zurückliefern. Benutzen Sie für die Implementierung den Laplace'schen Entwicklungssatz. Dieser besagt, dass für  $A = (a_{ij})$  gilt:

$$\det(A) = \sum_{i=1}^n (-1)^{i+1} \cdot a_{i1} \cdot \det(A_{i1})$$

wobei  $A_{i1}$  die  $(n-1) \times (n-1)$ -Untermatrix von  $A$  ist, die durch Streichen der  $i$ -ten Zeile und der ersten Spalte entsteht. Implementieren Sie diesen Algorithmus mit Hilfe einer Rekursion.

*Hinweis:* Für eine  $1 \times 1$  Matrix  $A = (a)$  gilt  $\det(A) = a$ .

## 3 Aufgabe: Connect Four

### 3.1 Ziele

- Vererbung vertiefen

### 3.2 Aufgabenstellung

Passen Sie ihre Klasse `Player` aus der Übung 7 so an, dass diese nun das folgende Interface implementiert:

Klasse `AbstractPlayer`

```
public interface PlayerInterface {  
  
    public boolean isFirst();  
  
    public int play(ConnectFour brett);  
}
```

Zusätzlich sollen Sie die Klasse `ConnectFour` anpassen, damit diese anstatt der `Player` Klasse das Interface `PlayerInterface` benutzt.

Nun können Sie mit Mitstudenten die Klassen `Player` austauschen. Ändern Sie dazu den Namen ihrer Klasse zu `Player_<name>`, wobei `<name>` Ihr Name ist. Nun kann ein Mitstudent beide Klassen in seinem Projekt verwenden und die beiden `Player` gegeneinander antreten lassen.