

# Programmierung II

Wintersemester 2021/22

Prof. Dr.-Ing. Marco Block-Berlitz

Prof. Dr.-Ing. Dietrich Kammer

Dipl.-Ing. Jan Roeper

# UML-Diagramme

## B01

Versuchen Sie ihr Lieblingsrezept in Pseudocode und anschließend in ein Aktivitätsdiagramm zu übertragen.

## B02

Überlegen Sie sich Fälle, bei denen ein Programm nicht terminiert.  
Verwenden Sie für die Erläuterung Aktivitätsdiagramme.

## B03

Erstelle ein Zustandsdiagramm zur Verwendung der Telefonapp auf dem Smartphone. Starten Sie beim Lockscreen über Homescreen bis zur Auswahl der Telefonapp. Beschreiben Sie die Verwendung der App mit einem Unterzustandsdiagramm (wählend, verbunden, usw.)

# JavaGotchi

## B04

Nehmen Sie sich das Projekt JavaGotchi noch einmal vor und geben Sie auch dort einen sauberen Entwurf an. Identifizieren Sie Funktionen und eine Programmschleife. Damit die Funktionen zu den jeweiligen Zuständen nicht so klein sind, erweitern Sie es. So könnte ihr Grundgerüst aussehen:

```
public class JavaGotchi {
    static enum State {HAPPY, HUNGRY, EATING, STARVED};
    static State currentState;
    static int hungerLevel;
    // ... maybe more functions
    static void behaviourHungry() { /* ... */ }
    static void behaviourStarved() { /* ... */ }
    static void behaviourEating() { /* ... */ }
    static void behaviourHappy() { /* ... */ }
    static void update() { /* ... */ }
    static void wait(int ms) { /* ... */ }
    static void startGame() {
        currentState = State.HAPPY;
        hungerLevel = 0;
        boolean isGameRunning = true;
        while (isGameRunning) {
            switch (currentState) {
                case HUNGRY:
                    behaviourHungry();
                    break;
                /* ... more cases */
            }
            update(); wait(400);
        }
    }
    public static void main(String[] args) { startGame(); }
}
```

# Programmparameter nutzen

## B05

Wir haben gesehen was passiert, wenn weniger Parameter übergeben werden als das Programm sie verlangt oder erwartet (Abschnitt 3.1). Testen Sie, wie das Programm reagiert, wenn Sie mehr als die erwartete Anzahl von Parametern übergeben.

```
public class MeineEingaben {  
    public static void main(String[] args) {  
        System.out.println("Eingabe 1: >"+args[0]+"< und");  
        System.out.println("Eingabe 2: >"+args[1]+"<");  
    }  
}
```

# Kodierung von Zeichen

B06

In Abschnitt 2.8.2 wurde folgender Programmabschnitt vorgestellt:

Ersetzen Sie die beiden Funktionen

`boolean Character.isUpperCase(char)`

`char Character.toLowerCase(char)`

durch eigene Implementierungen. Überlegen Sie sich weiterhin, wie Sie die Korrektheit bzw. Übereinstimmung der eigenen Funktionen mit den ersetzten überprüfen können.

Aussen:

```
for (int i = 33; i < 127; i++) {  
    for (int j = 33; j < 127; j++) {  
        char symbol1 = (char)i;  
        char symbol2 = (char)j;  
        if (Character.isUpperCase(symbol1) &&  
            (symbol2 == Character.toLowerCase(symbol1))) {  
            System.out.println("Symbol " + symbol1 + " [" + i +  
                               "]" und Symbol " + symbol2 + " [" + j +  
                               "]" sind ein Paar.");  
            continue Aussen;  
        }  
    }  
}
```

# Kryptographie

## B07

Bei der xor-Codierung hatten wir die folgende entscheidende Programmzeile:

```
zeichen[i] = (char)(zeichen[i]^key);
```

Erläutern Sie die Abarbeitung schrittweise.

## B08

Implementieren Sie das Verfahren der Caesarkodierung analog zur xor-Codierung. Auch hier können Sie versuchen, ein kleines Programm zu schreiben, dass einen vorliegenden, codierten, deutschen Text entschlüsselt.

# Mehr Zeichenkodierung

## B09

Versuchen Sie eine eigene Methode zu schreiben, die das häufigste Symbol in einer Zeichenkette bestimmt (analog zu `haeufigstesSymbol` im Projekt Codeknacker).

## B10

Die Codierung über Transpositionen (Verschiebungen) ist durch das Verfahren Skytale bereits seit dem 5. Jahrhundert v. Chr. bekannt. Dabei wurde ein Pergamentstreifen um einen speziellen Holzstab gewickelt und dort in Klartext über die Länge des Stabes eine Textnachricht geschrieben. Entrollt ergaben die Buchstabenteile keinen Sinn. Der Empfänger kann mit einem identischen Stab den Text wieder lesbar machen.

Überlegen Sie sich eine geeignete Möglichkeit, dieses Verfahren im Programmcode zu simulieren. Hinweis: Es wird zwischen den Textpassagen Platz vorhanden sein, füllen Sie diesen mit willkürlichem Text.