TU Kaiserslautern – FB Informatik
Dr. Patrick Michel

31.10.2019

Algorithmen und Datenstrukturen (WS 2019)

Aufgabenblatt 0

zu bearbeiten bis: 04.11.2019 / 06.11.2019

Das Fach AlgoDat baut auf zentralen Grundlagen aus *Objektorientierte Softwareentwicklung* auf, insbesondere Objektorientierung, Referenzen und Rekursion. Dieses Übungsblatt dient zur Wiederholung dieser Grundlagen und wird in der ersten Übung gemeinsam besprochen.

Aufgabe 0.1 (Referenzen - Theorie)

```java
public class Character {
    String name;
    House house;

    public Character(String name,
                     House house) {
        this.name = name;
        this.house = house;
    }



}
```

```java
public class House {
    String name;
    String words;

    public House(String name,
                 String words) {
        this.name = name;
        this.words = words;
    }

    public House clone() {
        return new House(this.name,
                         this.words);
    }
}
```

Mit den obigen beiden Klassen wird der folgende Code ausgeführt. Skizzieren Sie danach die Situation im Speicher: Welche Objekte existieren und wie referenzieren diese aufeinander?

```java
House stark = new House("Stark", "Winter is coming.");
House none = null;

Character john = new Character("John Snow", stark);
Character bran = new Character("Bran Stark", stark);
Character arya = new Character("Arya Stark", stark.clone());
Character tyrion = new Character("Tyrion Lannister", none);
```

Wir verpassen Johns Haus den Doppelnamen “Stark-Targaryen”. Was genau bewirken die folgenden Zeilen im Speicher? Was wird ausgegeben?

```
1  john.house.name += "-Targaryen";
2  System.out.println(bran.house.name);
3  System.out.println(arya.house.name);
```

Welche der folgenden Zeilen werfen eine Exception?

```
1  john.house.name = john.name;
2  tyrion.house.name = "Lannister";
3  House[] houses = new House[] {stark, stark, null};
```

Aufgabe 0.2 (Sortierung - Praxis)
Lösen Sie die folgenden Probleme jeweils mit einer iterativen Implementierung:

a) Gegeben ein Array von Integer-Werten, prüfen Sie ob das Array aufsteigend sortiert ist (Ergebnis `true`) oder nicht (Ergebnis `false`).

b) Gegeben ein Array von **Person**-Werten, prüfen Sie ob das Array aufsteigend nach dem Alter (`person.age()`) sortiert ist (Ergebnis `true`) oder nicht (Ergebnis `false`).

Aufgabe 0.3 (Exponentialfunktion - Praxis)
Lösen Sie die folgenden Probleme jeweils mit einer iterativen Implementierung:

a) Die Exponentialfunktion kann im Rechner approximiert werden, indem man die ersten $K+1$ Glieder der *Exponentialreihe* aufsummiert:

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} \approx \sum_{k=0}^{K} \frac{x^k}{k!}$$

Berechnen Sie diese Approximation für e^x. Achten Sie auf eine effiziente Implementierung.