

## Aufgabenblatt A05: Generics

### 1 Java Denksportaufgaben

1. Was gibt diese Program aus?

```
import java.util.*;
public class SuspiciousSort {
    public static void main(String[] args) {
        Random rnd = new Random();
        Integer[] arr = new Integer[100];
        for (int i = 0; i < arr.length; i++)
            arr[i] = rnd.nextInt();
        Comparator<Integer> cmp = new Comparator<Integer>() {
            @Override
            public int compare(Integer i1, Integer i2) {
                return i2 - i1;
            }
        };
        Arrays.sort(arr, cmp);
        System.out.println(order(arr));
    }
    enum Order {
        ASCENDING, DESCENDING, CONSTANT, UNORDERED
    };
    static Order order(Integer[] a) {
        boolean ascending = false;
        boolean descending = false;

        for (int i = 1; i < a.length; i++) {
            ascending |= a[i] > a[i - 1];
        }
    }
}
```

```

        descending |= a[i] < a[i - 1];
    }
    if (ascending && !descending)
        return Order.ASCENDING;
    if (descending && !ascending)
        return Order.DECENDING;
    if (!ascending)
        return Order.CONSTANT; // All elements equal
    return Order.UNORDERED; // Array is not sorted
}
}

```

- Wie kommt die Ausgabe zustande?
- Wie machen Sie das richtig?

2. Was gibt das folgende Programm aus? Ist es überhaupt compile-fähig? Was ist an diesem Code ggfs. verbesserungswürdig?

```

import java.util.Arrays;
import java.util.List;
public class Pair<T> {
    private final T first;
    private final T second;

    public Pair(T first, T second) {
        this.first = first;
        this.second = second;
    }

    public T first() {
        return first;
    }
    public T second() {
        return second;
    }
    public List<String> stringList() {
        return Arrays.asList(String.valueOf(first),
                               String.valueOf(second));
    }

    public static void main(String[] args) {
        Pair p = new Pair<Object>(23, "skidoo");
        for (String s : p.stringList())

```

```

        System.out.print(s + " ");
    }
}

```

3. Was ist mit dem folgenden Programm los? Ist etwas falsch? Wenn ja, was ist falsch und wie macht man das richtig? Wenn nein: Was tut es?

```

public class LinkedList<E> {
    private Node<E> head = null;

    private class Node<E> {
        E value;
        Node<E> next;

        // Node constructor links the node as a new head
        Node(E value) {
            this.value = value;
            this.next = head;
            head = this;
        }
    }

    public void add(E e) {
        new Node<E>(e);
        // Link node as new head
    }

    public void dump() {
        for (Node<E> n = head; n != null; n = n.next)
            System.out.println(n.value + " ");
    }

    public static void main(String[] args) {
        LinkedList<String> list = new LinkedList<String>();
        list.add("world");
        list.add("Hello");
        list.dump();
    }
}

```

## 2 Generische Klasse

Die Lernziele dieser Aufgabe sind:

- Ein Interface implementieren können.
- Auf Fehlersituationen sowohl mittels Return Code als auch Exception professionell reagieren können.
- Schreiben von Klassen weiter üben, auch mit Generics.

Schreiben Sie bitte eine generische Klasse mit folgenden Eigenschaften:

1. Sie implementiert das Interface Deque (siehe Datei zur Aufgabe im Lernraum).
2. Für die Speicherung wird eine einfach vorwärts verkettete Liste verwendet. Die Klasse *LinkedList* aus dem Paket *java.util* verwendet z. B. eine doppelt verkettete Liste. einen Versuch einer vorwärtsverketteten Liste sehen Sie in der letzten der drei Denksportaufgaben. Bei den hier formulierten Anforderungen haben Sie am Ende der Liste eine Referenz „über“. Ich denke es ist eine gute Idee, diese für eine Referenz auf das Element davor zu verwenden.
3. Implementieren Sie bitte auch hier Methoden wie *equals*, *hashCode* und *toString* professionell!

Testfälle für diese Aufgabe bestehen Sie auch, wenn Sie einfach eine Variable mit dem Typ des Interfaces deklarieren und mit einer *ArrayList* oder *LinkedList* implementieren. Der Sinn dieser Aufgabe ist aber, dass Sie eine generische Klasse selber implementieren lernen.

Der Abgabetermin für alle ist:

**Mittwoch, 06.05.2020, 08:00**