Ibraheem Saoud (3041390)
Liang Chun Wong (3037494)
Miroslav Valov (3023792)

Aufgabe 8.1

(1) Type

QUEUE[T]

(2) Function

mt_queue: QUEUE[T]

enqueue: $T \times QUEUE[T] \to QUEUE[T]$

dequeue: $QUEUE[T] \nrightarrow QUEUE[T]$

head: $QUEUE[T] \nrightarrow T$

empty: $QUEUE[T] \to Bool$

(3) Preconditions

$\forall$ q: QUEUE[T] $\bullet$

pre(dequeue(q)) $\Leftrightarrow$ empty(q) = false

pre(head(q)) $\Leftrightarrow$ empty(q) = false

(4) Axioms

$\forall$ x, y : T; q: QUEUE[T] $\bullet$

dequeue(enqueue(x, mt_queue)) = mt_queue

q $\neq$ mt_queue $\Rightarrow$ dequeue(enqueue(x, q))) = enqueue(x, dequeue(q)))

head(enqueue(x, mt_queue)) = x

q $\neq$ mt_queue $\Rightarrow$ head(enqueue(x, q))) = head(q))

empty(mt_queue) = true

empty(enqueue(x, q)) = false

Aufgabe 8.2

```java
// Array.java
public interface Array<T> {
    void put(int pos, T x);
    int lower();
    int upper();
    T get(int pos);
    boolean empty();
}
```

Ibraheem Saoud (3041390)
Liang Chun Wong (3037494)
Miroslav Valov (3023792)

```java
// MyArray.java
import java.util.ArrayList;

public class MyArray<T> implements Array<T> {
    public ArrayList<T> arr;
    public int i;
    public int j;

    MyArray(int i, int j) {
        this.arr = new ArrayList<T>(j);
        for (int x = 0; x < i; x++) {
            this.arr.add(null);
        }
        for (int x = i; x <= j; x++) {
            this.arr.add(null);
        }
        this.i = i;
        this.j = j;
    }

    public int lower() {
        return this.i;
    }

    public int upper() {
        return this.j;
    }

    public T get(int pos) {
        if (pos < this.i) return null;
        return this.arr.get(pos);
    }

    public boolean empty() {
        for (int i = 0; i <= this.j; i++) {
          if(this.arr.get(i) != null)
             return false;
        }
        return true;
    }

    public void put(int i, T x) {
        if (i > this.j || i < this.i) return;
        this.arr.set(i, x);
    }

    public String toString() {
        ArrayList<T> clone = (ArrayList<T>)this.arr.clone();
        for (int i = this.i - 1; i >= 0; i--) {
            clone.remove(i);
        }
        return clone.toString();
    }
}
```

```java
// Main.java
public class Main {
    public static void main(String args[]) {
        // Create Array[T] with type String and bounds 2 to 5
        MyArray<String> x = new MyArray<>(2, 5);

        // Insert two elements
        x.put(2, "ONE");
        x.put(3, "TWO");

        // Print the array
        System.out.println(x);

        // Check if it's empty
        System.out.println(x.empty());

        // Get element at position two
        System.out.println(x.get(2));

        // Get lower and upper bounds
        System.out.println("Lower bounds: " + x.lower());
        System.out.println("Upper bounds: " + x.upper());

        // Out of range put
        x.put(6, "INVALID ELEMENT");
        // Nothing will show in this println statement
        System.out.println(x);

    }
}
```

Aufgabe 8.3

```
8.3.1 Bubble Sort:
start a: [3, 1, 6, 7, 2, 5, 4]
1. loop: [1, 3, 6, 2, 5, 4, 7]
2. loop: [1, 3, 2, 5, 4, 6, 7]
3. loop: [1, 2, 3, 4, 5, 6, 7]
4. loop: [1, 2, 3, 4, 5, 6, 7]

8.3.2 Insertion Sort:
start a: [3, 1, 6, 7, 2, 5, 4]
1. loop: [1, 3, 6, 7, 2, 5, 4]
2. loop: [1, 3, 6, 7, 2, 5, 4]
3. loop: [1, 3, 6, 7, 2, 5, 4]
4. loop: [1, 2, 3, 6, 7, 5, 4]
5. loop: [1, 2, 3, 5, 6, 7, 4]
6. loop: [1, 2, 3, 4, 5, 6, 7]

8.3.3 Selection Sort:
start a: [3, 1, 6, 7, 2, 5, 4]
1. loop: [1, 3, 6, 7, 2, 5, 4]
2. loop: [1, 2, 6, 7, 3, 5, 4]
3. loop: [1, 2, 3, 7, 6, 5, 4]
4. loop: [1, 2, 3, 4, 6, 5, 7]
5. loop: [1, 2, 3, 4, 5, 6, 7]
6. loop: [1, 2, 3, 4, 5, 6, 7]
```