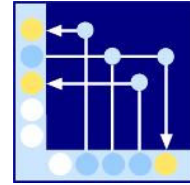




Hochschule Aalen

*Fakultät Elektronik und Informatik
Studienbereich Informatik*



Objektorientierte Programmierung

Vorlesung im Wintersemester 2023/2024

Prof. Dr. habil. Christian Heinlein

10. Übungsblatt (18. – 24. Januar 2024)

Aufgabe 10: Ausnahmen

Teilaufgabe 10.a)

Definieren Sie eine Java-Schnittstelle `Queue` zur Beschreibung einer Warteschlange für Zeichenketten mit folgenden Methoden:

- `size` liefert die Größe der Schlange, d. h. die Anzahl der momentan gespeicherten Elemente.
- `elem` liefert das `i`-te Element der Schlange, wenn der Parameter `i` zwischen 0 einschließlich und der Größe der Schlange ausschließlich liegt; andernfalls wird eine Ausnahme des Standardtyps `IndexOutOfBoundsException` geworfen, deren Konstruktor den fehlerhaften Indexwert als Zeichenkette erhält.
- `enqueue` fügt ein Element am Ende der Schlange hinzu.
Wenn `null` übergeben wird, wird eine Ausnahme des Standardtyps `NullPointerException` geworfen.
- `dequeue` liefert und entfernt das erste Element der Schlange.
Wenn die Schlange momentan leer ist, wird eine geprüfte Ausnahme des Typs `QueueEmptyException` geworfen.

Teilaufgabe 10.b)

Definieren Sie die in Teilaufgabe a erwähnte Ausnahmeklasse `QueueEmptyException` mit einer öffentlichen, unveränderlichen Objektvariablen des Typs `Queue` sowie einem öffentlichen Konstruktor mit entsprechendem Parameter zu ihrer Initialisierung!

Teilaufgabe 10.c)

Schreiben Sie eine Klasse `ListQueue`, die die Schnittstelle `Queue` mit Hilfe einer (selbst implementierten) verketteten Liste implementiert! Der parameterlose öffentliche Konstruktor der Klasse soll eine leere Warteschlange erzeugen. Alle Datenfelder der Klasse sollen privat sein.

Teilaufgabe 10.d)

Ergänzen Sie im folgenden Testprogramm `catch`-Klauseln für alle geprüften und ungeprüften Ausnahmen, die innerhalb des `try`-Blocks auftreten können! Erläutern Sie jeweils durch Kommentare, unter welchen Umständen die jeweilige Ausnahme auftreten kann, und geben Sie jeweils eine passende Fehlermeldung aus! Denken Sie auch an Ausnahmen, die von Operatoren und Bibliotheksmethoden geworfen werden können!

```
class QueueTest {
    public static void main (String [] args) {
        // Der Code des try-Blocks darf nicht verändert werden!
        try {
            // Warteschlange q erzeugen.
            Queue q = new ListQueue();

            // Kommandozeilenargumente verarbeiten.
            for (String a : args) {
                // "-" bedeutet: erstes Element von q entfernen und ausgeben.
                if (a.equals("-")) {
                    System.out.println(q.dequeue());
                }
                // "@NNN" bedeutet: NNN-tes Element von q abfragen und ausgeben.
                else if (a.charAt(0) == '@') {
                    int j = Integer.parseInt(a.substring(1));
                    System.out.println(q.elem(j));
                }
                // Andere Argumente bedeuten: a am Ende von q anfügen.
                else {
                    q.enqueue(a);
                }
            }
            // Hier sollen catch-Klauseln ergänzt werden!
        }
    }
}
```

Hinweis

Die Dokumentation von Java-Bibliotheksklassen wie z.B. `Integer` und `String` (aus dem Paket `java.lang` im Modul `java.base`) findet man online z.B. unter docs.oracle.com/javase/21/docs/api → `java.base` → `java.lang`.