Kapitel 6: Exceptions

Lernziele

[LZ 6.1] Das Java-Konzept zur Fehlerbehandlung kennen und das Prinzip erklären können
[LZ 6.2] Die Klassenhierarchie zur Fehlerbehandlung kennen und erläutern können
[LZ 6.3] Den Unterschied zwischen *checked* und *unchecked* Exceptions erklären können
[LZ 6.4] Die beiden Arten des Umgangs mit Exceptions erklären und anwenden können
[LZ 6.5] Eigene Exception-Klassen definieren und nutzen können
[LZ 6.6] Den Sinn des finally-Blocks erklären können und finally-Blöcke anwenden können

Erläutern Sie das Java-Konzept zur Fehlerbehandlung

6. Exceptions

[LZ 6.2] Die Klassenhierarchie zur Fehlerbehandlung kennen und erläutern können

- Welche Klassen gibt es in Java im Zusammenhang mit Fehlerbehandlung und wozu werden diese genutzt?

6. Exceptions

[LZ 6.3] Den Unterschied zwischen checked und unchecked Exceptions erklären können

- Was ist eine checked Exception?
- Was ist eine unchecked Exception?
- Wann wird welche verwendet, welche verwendet man bei der Anwendungsentwicklung typischerweise?

6. Exceptions

[LZ 6.4] Die beiden Arten des Umgangs mit Exceptions erklären und anwenden können

- Wie kann man mit Exceptions umgehen?
- Geben Sie für jede der beiden Arten des Umgangs mit Exception ein Beispiel an!

- Definieren Sie eine Exception Klasse zur Speicherung folgender Daten:
- Stacktrace
- Zeitstempel des Fehlers
- Meldung für Benutzer
- E-Mail-Adresse des zuständigen Systemadministrators
- Fehlermeldung
- Definieren Sie eine Methode, in der sie obige Exception werfen. Die Exception soll in der main-Methode abgefangen werden und zur Ausgabe aller Daten der Exception führen
- Definieren Sie eine weitere Methode, in der sie obige Exception werfen. Die Exception soll nun lokal abgefangen werden und zur Ausgabe aller Daten der Exception führen
- Definieren Sie eine main-Methode, in der Sie beide Methoden aufrufen

[LZ 6.6] Den Sinn des finally-Blocks erklären können und finally-Blöcke anwenden können

- Wofür werden finally-Blöcke verwendet?
- Geben Sie ein Beispiel einer void-Methode an, die einen finally-Block definiert und rufen Sie diese von main aus auf.
- Geben Sie ein Beispiel einer int-Methode an, die einen finally-Block definiert und rufen Sie diese von main aus auf. Was stellen Sie bzgl. des Methodenergebnisses fest?

Übung (Prüfungslevel)

```
class StringListe {
    private String[] strings = new String [10];
    void einfuegen(int index, String element) throws ArrayIndexOutOfBoundsException {
        // a)...
        strings[index] = element;
        // b)...
    }
    String gibStringZuIndex(int index) {
        if (index < 0 || index > strings.length-1)
            return null;
        else
            return strings[index];
    }

    public static void main(String[] args) throws Exception {
        StringListe l = new StringListe() ;
        l.einfuegen(6, "sieben");
        l.einfuegen(20, "einundzwanzig");
        System.out.println(l.gibStringZuIndex(20));
    }
}
```

Ergänzen Sie die Stellen a) und b) so, dass eine ArrayIndexOutOfBoundsException abgefangen wird. Ist die Ursache ein positiver, zu großer Index, so soll die Reihung entsprechend vergrößert werden und der Zugriff anschließend durchgeführt werden.