Tutorium zur Vorlesung Programmieren

9. Tutorium

Annika Berger

16. Januar 2017

IPD Koziolek

Magic Numbers

Magic Numbers

Zahlen im Code ohne Erklärung

Tile.java

```java
for (int i = 0; i < 6; i++) {
    ...
}
```

Board.java

```java
@Override
public String toString() {
    final StringBuffer result = new StringBuffer();
    for (int i = 0; i < this.tiles.length; i++) {
        result.append(this.tiles[i]);
        result.append(';');
        if ((i + 1) % 3 == 0) {
            result.append(System.lineSeparator());
        }
    }
    return result.toString();
}
```

Magic Numbers

Zahlen im Code ohne Erklärung

Tile.java

```
for (int i = 0; i < lineTypes.length; i++) {
    ...
}
```

Board.java

```
private static final int COLUMN_LENGTH = 3;
@Override
public String toString() {
    final StringBuffer result = new StringBuffer();
    for (int i = 0; i < this.tiles.length; i++) {
        result.append(this.tiles[i]).append(';');
        if ((i + 1) % COLUMN_LENGTH == 0) {
            result.append(System.lineSeparator());
        }
    }
    return result.toString();
}
```

Exceptions

Exceptions

Behandeln eine Fehlersituation (Ausnahme) im Programmablauf

- Erzeugung mit **new**
- Auslösen („Werfen“) mit **throw**
- Immer aussagekräftige Fehlermeldung mitgeben

java.util.ArrayList (OpenJDK 8)

```
659 private void rangeCheckForAdd(int index) {
660     if (index > size || index < 0)
661         throw new
              ↪  IndexOutOfBoundsException(outOfBoundsMsg(index));
662     }
663 }
```

try – catch

Fehlerbehandlung mit try-catch-Blöcken

```
try {
    // code in dem eine BadExampleException auftreten kann
} catch (BadExampleException badException) {
    // Behandlung durch andere Methode aufrufen, Fehlermeldung
      ↪  ausgeben, o.ä.
    System.out.println(badException.getMessage());
}
```

The Tutor cannot think of a better Example.

try – catch

AbstractList.java

```
 public void add(E e) {
     checkForComodification();
     try {
         int i = cursor;
         AbstractList.this.add(i, e);
         lastRet = -1;
         cursor = i + 1;
         expectedModCount = modCount;
     } catch (IndexOutOfBoundsException ex) {
         throw new ConcurrentModificationException();
     }
}
```

throws

Wird die **throws**-Klausel zur Signatur hinzugefügt, wird die Fehlerbehandlung an die aufrufende Methode weitergegeben.

java.util.ArrayList (OpenJDK 8)

```
774 private void readObject(java.io.ObjectInputStream s) throws
      ↪  java.io.IOException, ClassNotFoundException {
775     ...
776 }
```

RuntimeExceptionS

- Zeigen Fehler des Programmierers auf
- nicht behandeln, sondern beheben

Eigene Exceptions erstellen

Unterklasse von `Exception` oder `RuntimeException`

Verboten

- `try`-Block um das gesamte Programm
- Leerer `catch` Block
- Fangen vom Typ `Exception` & `Throwable`
- Grundsätzlich überall Exceptions einfügen

Übung

Präsenzübung

- Keine Hilfsmittel zugelassen; am Tisch nur: dokumentenechter Stift, Studierendenausweis und Personalausweis
- Aufgabenblätter liegen lassen bis Startsignal kommt
- Nach Abgabe warten bis Abgaben durchgezählt sind
- Jeder Betrugsversuch sofort „nicht bestanden“

Übungsaufgaben

judge.joshuagleitze.de

- Aufgaben sind im DOMJudge hinterlegt
- Prüft eure Abgabe, bevor ihr sie hochladet