II.4.4 Exceptions

Dienstag, 5. Dezember 2017 09:00

- Explizites Abfragen aller möglichen Fehlerquellen
  → ineffiziente und unleserliche Programme
- Exceptions treten automatisch bei entspr. Fehler auf.
  Beim Auftritt einer Exception kann diese in einem eigenen Programmstück behandelt werden (Exception Handler).
  Fällt dieses Programmstück, dann bricht Prog. mit Fehlermeldung ab.
- Exception Handler

```
try {....}
catch (...) {...}
      :
catch (...) {...}
finally {....}
```

Man versucht, diesen Block auszuführen.
Wenn Ausnahme "geworfen" wird, dann wird sie im passenden "catch" gefangen und der entspr. catch-Block ausgeführt. Je nach Typ des Exc.-Objekts wird der richtige catch-Block gewählt.

- Idee: Error sind Feller, von denen sich das Prog. i.A. nicht erholen kann, sondern die zum Abbruch führen sollten.
- Unchecked exceptions:
  alle Objekte der Klasse Error und RuntimeException

  Diese Exceptions dürfen gefangen werden, muss man aber nicht.
- Alle anderen Exceptions sind checked exceptions, die gefangen werden müssen (sonst Compile-Fehler).
- Exceptions werden bei Fehlern automat. erzeugt. Man kann sie aber auch explizit selbst erzeugen (mit "new" und Konstruktor).
  String-Arg. könnte Information über die Exc. enthalten.

Eigene Implementierung v. Exception-Klassen

- Unterklassen v. Throwable

- Bsp: NegativeNumberExc. hat ein Attribut, das die jeweilige negative Zahl enthält.
- Explizites Werfen v. Exceptions: mit "throw"
- Bsp-Ausgabe:

falls x=3: Fakultät von 3 ist 6.

falls x=-3: Fehler! -3 < 0

- Wenn in einer Methode checked Exceptions auftreten können die nicht gefangen werden, dann muss dies im Methoden-Kopf mit "throws" deklariert werden.
- Javadoc hat einen Tag @exception zur Beschreibung der Exceptions, die eine Methode werfen kann.
- Es kann mehrere catch-Statements geben. Dann wird die catch-Anweisung mit dem speziellsten passenden Parameter-typ ausgeführt. Dann müssen catch-Anweisungen mit speziellen Parameter-Typen vor catch-Anweisungen mit allgemeinen Parameter-Typen kommen.
- Bsp-Ausführung:

bei x=17: Fehler! Es trat die folgende Ausnahme auf:
17 ist zu groß.
↑
Durch die toString-Methode v. TooBigNumberExc.

- Bsp-Ausgabe mit "finally":

bei: x=3: Fak. von 3 ist 6
Ende des try-catch Blocks

Ende der Methode test

bei x = -3: Fehler! $-3 < 0$
Ende des try-catch Blocks
Ende der Methode test

bei x = 17: Ende des try-catch Blocks
Fehler! Es trat die folgende Ausnahme auf:
17 ist zu groß

Exceptions sind nur für Ausnahmebehandlung gedacht, nicht für normale Verzweigungen oder Sprünge.