

- Exceptions werden automatisch "geworfen" – oft einfacher, als jede Fehlermöglichkeit vorher abzufragen.
- Exceptions können vom Programm "gefangen" werden u. entsprechend behandelt werden. Ansonsten endet es mit Programmabsturz.

=> try...catch-Blöcke

- Für verschiedene Typen v. Exceptions kann es unterschiedliche catch-Blöcke geben. Diese unterscheiden sich nach dem Typ der Exception.

### Arten von Ausnahmen

- Error: Fehler, von denen sich das Prog. normalerweise nicht erholt.

müssen nicht gefangen werden

- Exceptions :

Runtime Exceptions: sehr oft auftretende Exceptions  
müssen nicht gefangen werden

Alle anderen Exceptions (die U-Klasse v. Exception, aber nicht v. RuntimeException sind) müssen gefangen werden:  
checked exceptions

### Arbeitsweise von try-catch

Bsp.: A und B sind keine U-Klassen voneinander.

Beides sind unchecked Exceptions.

Wenn eine Exception geworfen wird und sie kann in der eigenen Methode nicht gefangen werden, dann wird die Methode abgebrochen und man springt an die Stelle zurück, an der die Methode aufgerufen wurde.

0  
Methode aufgerufen wurde.

Kann die Exception nirgendwo gefangen werden, bricht das Programm ab.

## Eigene Exception-Klassen

- Viele Exceptions werden automatisch geworfen.  
Aber man kann Exceptions auch selbst explizit werfen: throw
- Wenn eine Methode eine checked Exception nicht selbst fängt, die während ihrer Ausführung auftreten könnte, dann muss dies im Methodenkopf angegeben werden: throws

Bsp:

$x=3$ : Fakultät von 3 ist 6

$x=-3$ : Fehler!  $-3 < 0$ .

## Exception-Klassenhierarchie

- Mehrere catch-Statements werden von oben nach unten abgearbeitet.  
Es wird die erste passende catch-Anweisung ausgeführt.
- Man darf nicht erst Exceptions der Oberklasse mit catch fangen und dann Exceptions der U-Klasse (wird v. Compiler überprüft).

Bsp:

$x=-3$ : Fehler!  $-3 < 0$ .

$x=13$ : Fehler! Es tritt die folgende Ausnahme auf: 13 ist zu groß

↑  
gäbe sonst Überlauf

↑  
toString()  
von TooBigNumberExc.

finally: wird immer zum Schluss des try-catch-Blocks ausgeführt.

Bsp. mit test():

Bsp. mit `test()`:

- $x=3$ : Fak. von 3 ist 6  
Ende des try-catch-Blocks.  
Ende der Methode `test`.
- $x=-3$ : Fehler!  $-3 < 0$ .  
Ende des try-catch-Blocks.  
Ende d. Methode `test`.
- $x=13$ : Ende des try-catch-Blocks.  
Fehler! Es trat die folgende Ausnahme auf: 13 ist zu groß.

Generell: Verwende Ausnahmen sinnvoll

- nicht für Sprünge
- nicht für Fallunterscheidungen