Lesson 08.05.2018

Interfaces und statische Methoden



getZahl dann nur über Interface I aufrufbar



//Java8 -> Lambda Methoden mit functional Interfaces

Exceptions



Throwable

Checked

Unchecked: E die auftauchen können aber nicht behandelt werden müssen

Zeigwas:



Was passiert: NullPointer exception

Behandlung: try catch



Oder:



Andere Exceptions:



ArrayIndexOutOfBoundsExceptions kommt hier nicht zum tragen

Nur dann z.Bsp



Bzw:



Aufteilung in verschiedene catch-Blöcke

Beispiel: fileInputStream, Lesender Zugriff auf eine bestimmte Datei



Unsuported/Unreported Exception  
Diese ist checked: d.h. ich muss diese behandeln

* Mit catch



Was ist die Elternklasse von fileNotFound -> javadoc

Bsp IO



Weitere Verwendung



* 2. CatchBlock für IOException

Bei Tausch der Blöcke: Fehler, da Exceptions nacheinander abgearbeitet werden.

* Angabe der E in best. Reihenfolge: spezielle zuerst, generelle nachfolgend.

Weglassen des Try-Blocks: NetBeans: Handle Exception, Deklarieren

Throws Behandlung

Teil2

Exceptiones auslagern:



* Throws E



* Muss für den Aufruf dann static def werden (hier mit Aufruf aus main)



* Mit try catch:



Oder undifferenzierter/allgemeiner: -> Exception



Verwendung von Logging:







E auf jeder beliebigen Stufe behandelbar.



Handle machen:



Error wenn keine Quelle für Exception-Behandlung vorhanden

? Strukturierung von mehreren Error-Behandlungen, z.Bsp. Zusammenlegung

Bsp:



Unterschied checked und unchecked

* Unchecked: muss nicht behandelt werden
* Checked: ja

Execeptions werden behandelt Errors nicht !! (Tabelle)

Bsp. NullPointer E (ist unchecked)

Veränderen auf Checked E.



Beispiel: OutOfBounds



* Setzten von try catch



Runtime E

Fehler provozieren: realLocalData(25); ->



Folie

* Mehrere behandeln -> via Neuzweisung
* Multicatch (mehrere Exceptiones müssen vom unterschiedlichen Typ sein)

Beispiel: Neuzuweisung eines Catch Blocks  


Muticatch: Zusammenfassung von besonderen Exceptions.



Folie: …letzte Amtshandlung

Finally - wird immer ausgeführt



Methode wird abgebrochen und es geht zu main



Try mit finally geht:



Try Catch ohne finally nicht

Besondere Art des Trys seit java7



.. muss wieder geschlossen werden:



Schließen/Close kann wiederum Exception verursachen die abgefangen werden muss.



Neu Art seit Java7

  
Schließen jetzt nicht mehr nötig -> AutoCloseable   
Verbindung wird geschlossen wenn try verlassen wird.

Einbau von AutCloseable  


Folie: Exemtion weitergeben

Rethrow

Folie: Eigene Exceptiones