Exceptions & Errors

Java Foundation Track by Carsten Bokeloh

Exceptions & Errors

- Was ist eine Exception
- Abfangen von Exceptions
- RuntimeException
- Definieren eigener Exceptions
- Vererbung & Exceptions
- finally
- Throwable & Error

Was ist eine Exception

- Exception = Ausnahme
- eine Situation, die normalerweise im Programm nicht auftreten sollte
- zur Behandlung wurde die Klasse java.lang Exception entwickelt
- alle Exceptions erben von java.lang.Exception
- arithmetische Exceptions werden durch ArithmeticException behandelt.

Was ist eine Exception

- Zugriff auf Fehlermeldung durch excep.getMessage()
- StackTrace gibt Auskunft über Exception

Was ist eine Exception

Klassenname

Message

```
Exception in thread "main"
java.lang.ArithmeticException: / by zero
  at exceptions.Excep1.main(Excep1.java:9)
```

Ort des Auftretens

Abfangen von Exceptions

- Exception entsteht und wird „geworfen“ (**throw**)
- das Gegenstück zum Werfen ist Fangen (**catch**)
- Divison durch 0 soll unser Programm nicht zum Absturz bringen
  - wir fangen die Exception durch den try catch Block

Abfangen von Exceptions

```
package exceptions;

import java.io.FileReader;

public class Excep2 {

    public static void main(String arg[]) {
        FileReader fileReader = new FileReader("test.txt");
    }

}
```

Abfangen von Exceptions

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
        Unhandled exception type FileNotFoundException

        at exceptions.Excep2.main(Excep2.java:9)
```

Fangen von Exceptions

- Ausnahmesituation muss behandelt werden
  - aber wie?

Abfangen von Exceptions

- Lösung ist:
  - Weiterwerfen oder
  - Fangen und behandeln

Und warum da nicht?

```
package exceptions;

import Prog1Tools.IOTools;

public class Excep1 {
    public static void main(String arg[]) {
        int a = IOTools.readInteger("a=");
        int b = IOTools.readInteger("b=");
        System.out.println("a/b=" + (a / b));
    }

}
```

zwei Arten von Exceptions

- Java unterscheidet zwei Arten von Exceptions
  - „gewöhnliche“ Exception **abgeleitet von Exception**
  - RuntimeException **abgeleitet von Exception**

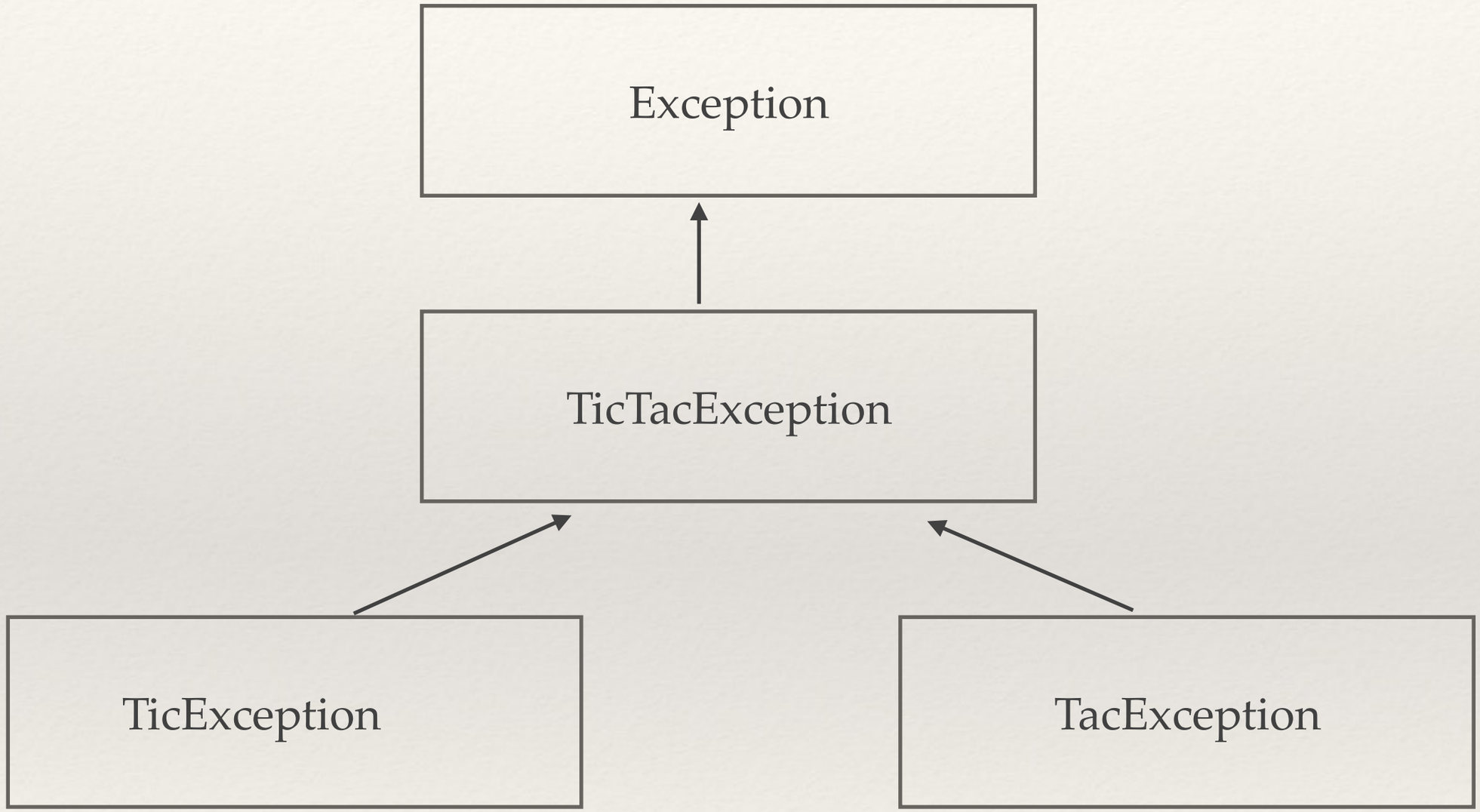
RuntimeException

- werden in Java hauptsächlich dazu verwendet, besonders häufig auftretende Ausnahmesituationen zu modellieren.
- diese jedesmal zu behandeln wäre zu aufwendig

Definieren eigner Exceptions

- public class EigeneException extends RuntimeException
- Konstruktor überschreiben für eigene Message

Vererbung & Exceptions



Abfangen von Exceptions

1. Stufe

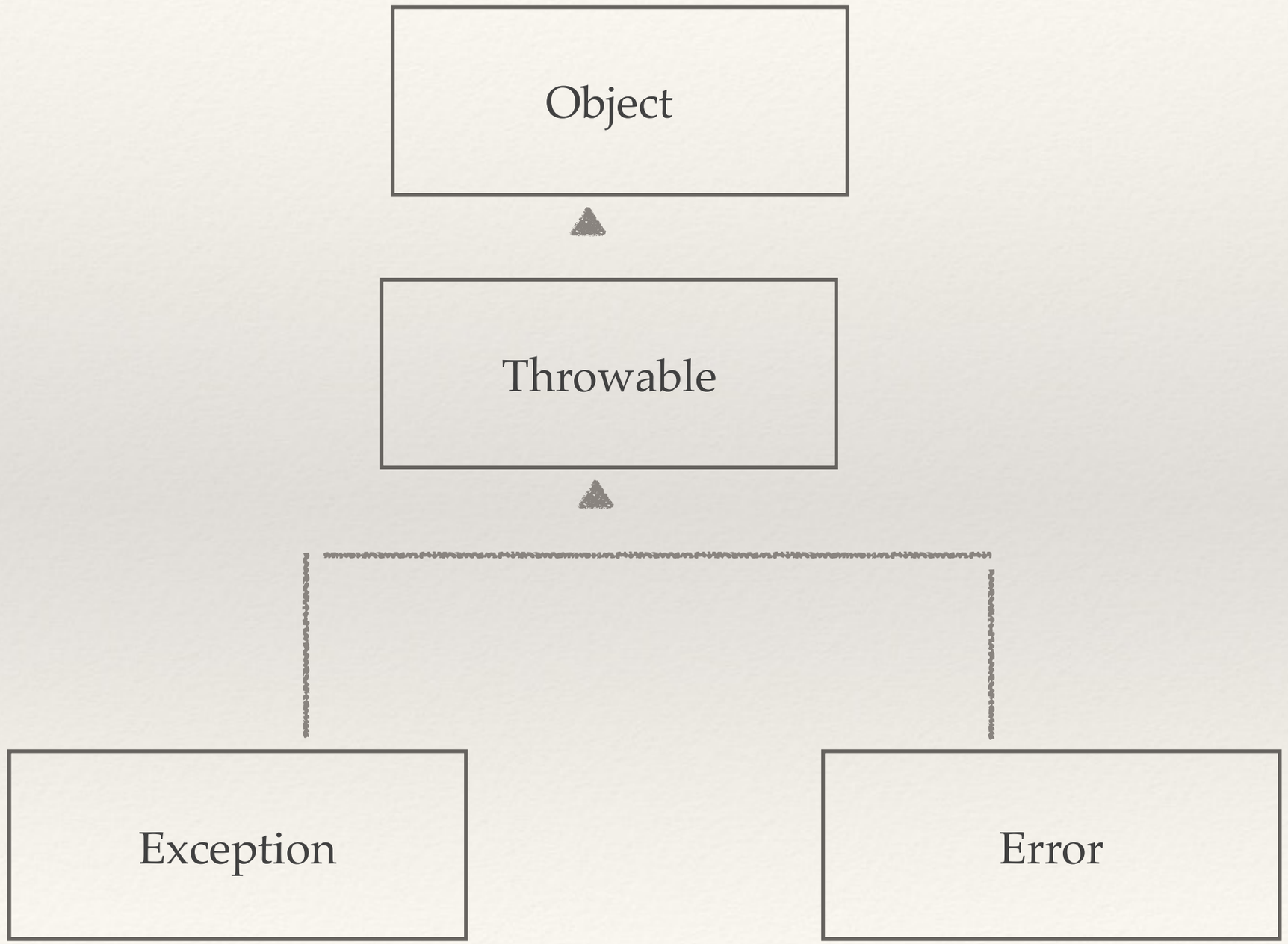
2.Stufe

3.Stufe

finally

- Was aber auf jeden Fall geschehen soll
  - steht im finally

Throwable & Error



Error

- stellen normalerweise schwerwiegende Ausnahmesituationen dar
  - NoClassDefFoundError

Assertions

- Zusicherungen im Programmcode
- assert <<Ausdruck>>;
- z.B: assert x != 0

Im Buch gibt es mehr Infos unter Kapitel 6.1

http://openbook.galileocomputing.de/javainsel/javainsel_06_001.html#dodtp97490f9b-54f8-4fd3-a4d6-482f5243e8ee

Übung

- Was ist korrekt?
  - Man kann keine RuntimeExceptions behandeln.
  - Man sollte keine Errors behandeln
  - Wenn eine Methode eine checked Exception wirft, muss Sie entweder von der Methode behandelt werden oder in der throws Anweisung weitergeworfen werden
  - RuntimeExceptions sind CheckedExceptions

Übung

```
class TryFinally {
    int a = 10;
    try {
        ++a;
    }
    finally {
        a++;
    }
    return a;
}
```

a: 10

b: 11

c: 12

d: Compilation Error

e: Runtime Exception

Übung

Definieren Sie sich eine eigene Exception ZufallszahlException, die von Exception erbt.
Schreiben Sie eine Klasse ExceptionUebung. Diese Klasse besitzt eine statische Methode gibZufallsZahlBisEinhalb.
In dieser Methode wird über java.lang.Math.random eine Zufallszahl generiert und zurückgegeben.
Ergibt Math.random() eine Zahl, die größer 0.5 ist, schmeißen Sie Ihre selbstdefinierte ZufallszahlException mit der Message "Zahl zu gross".
Die main Methode von ExceptionUebung ruft diese statische Methode auf. Für den Fall, dass eine Exception ausgelöst wird, soll diese behandelt werden, indem 0.5 ausgeben wird.
Im "normalen" Fall wird die Zufallszahl ausgegeben.