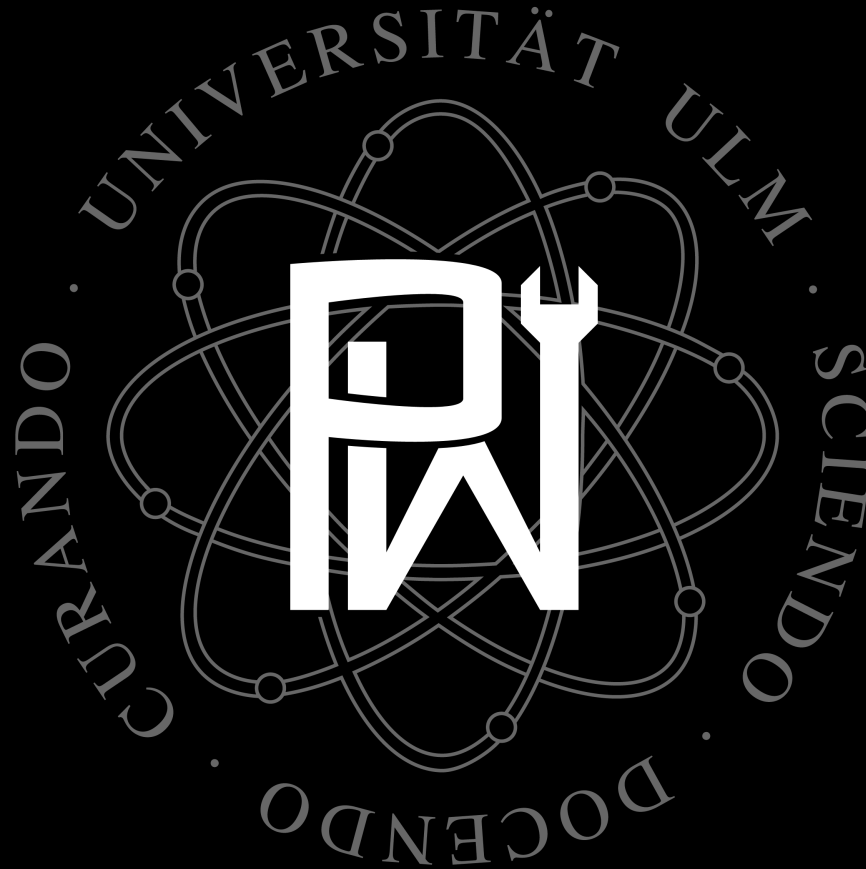


Programmierwerkstatt



Einführung in Java



Motivation

- Java ist leichter und übersichtlicher als C
- Keine Zeiger mehr
- gute Entwicklungsplattformen
- Mit Java kann man alles programmieren:
Handy, Web, Windows + Linux + Mac



Parameterübergabe bei Methoden

- Zwei verschiedene Arten der Parameterübergabe
- Abhängig vom Datentyp
- Call by Value
 - *Speicherbereiche von aufrufender und aufgerufener Methode sind voneinander getrennt. (int, float, double, ...)*
- Call by Reference
 - *Aufrufende und aufgerufene Methode greifen auf selben Speicherbereich zu. (Arrays, Listen, Objekte)*



Packages

- Klassen können in Packages zusammengefasst werden
- Dient der Strukturierung
- Momentan beinhaltet die Java-Klassenbibliothek 140 Packages mit über 2400 Klassen
- Die package-Anweisung legt fest, zu welchem Package die Klasse gehört



Import

- In einem Java-Programm können schon bestehende Klassen wieder verwendet werden
- Die import-Anweisung sagt dem Compiler, welche Klassen genutzt werden

```
import java.util.Calendar;
```

package Klasse

- Es können auch alle Klassen aus einem Package genutzt werden
- Hier werden nur die kompiliert, die benötigt werden



Import Beispiel

```
import java.util.Calendar;
```

```
// bzw:
```

```
import java.util.*;
```

```
public class ImportBeispiel {
```

```
    public static void main (String args[]) {
```

```
        Calendar c = Calendar.getInstance();
```

```
        System.out.println(c.get(Calendar.DATE)+". "+
```

```
(c.get(Calendar.MONTH)+1)+". "+c.get(Calendar.YEAR));
```

```
    }
```

```
}
```

Das System Objekt

- Objekt ist die Schnittstelle zum Host-System
- `System.out.println()`
- `System.in`

```
public static String readString() {  
    try {  
        BufferedReader in = new BufferedReader(  
            new InputStreamReader(System.in) );  
        String s = in.readLine();  
        return s; // Gelesenen String zurückgeben  
    } catch( IOException ex ) {  
        ex.printStackTrace();  
    }  
    return ""; //- Nichts gelesen -  
}
```

- `System.currentTimeMillis()`
- `System.exit(EXITCODE)`

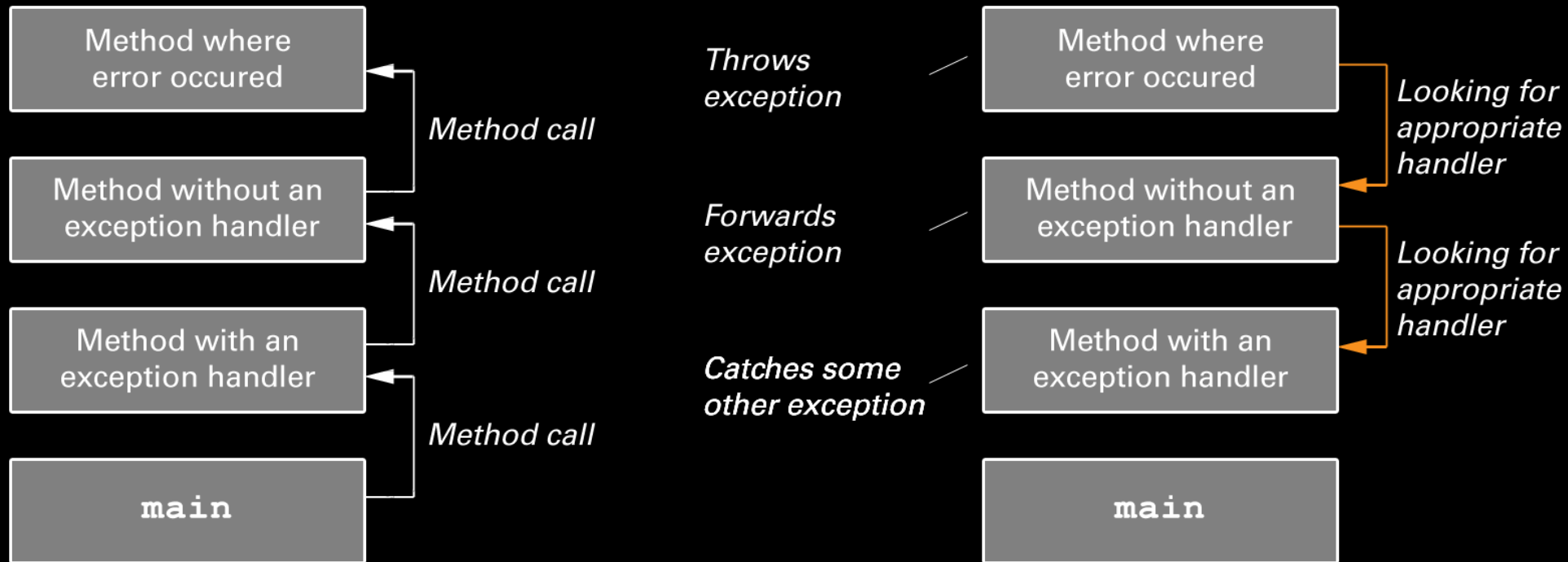


Exceptions

- Fehler treten definitiv irgendwann auf
- Wie umgehen mit Fehlern?
- In Java: Exceptions
- Definition:
"An exception is an event, which occurs during the execution of a program, that disrupts the normal flow of the program's instructions.,,"

→ Ereignis, das den normalen Programmablauf unterbricht

Ablauf von Exceptions





Umgang mit Exceptions

- **try/catch**
 - `try`-Block: Methoden, die evtl. Exceptions auslösen
 - `catch`-Block: Reaktion auf Exception
 - auch mehrere `catch`-Blöcke möglich
 - `finally`: Aufruf, nachdem `try`-Block verlassen wurde, egal auf welche Art (optional)
- **throws**
 - Exception wird an nächste Methode im Call-Stack weitergegeben
- **entweder "try/catch" oder "throws" muss verwendet werden**
 - Ausnahme:
RuntimeException (kann behandelt werden, muss aber nicht)



Links

- Skripte, Bücher und Tutorials
 - <http://www.mathematik.uni-ulm.de/sai/ss06/sep/>
 - <http://www.galileocomputing.de/openbook/javainsel4/>
 - <http://www.javabuch.de>
 - <http://java.sun.com/docs/books/tutorial/>
- Javadoc
 - <http://java.sun.com/j2se/1.5.0/docs/api>

Zum Schluss...

- Fragt, wenn euch etwas unklar ist
→ Wir werden schließlich dafür bezahlt!
- Erreichbar sind wir
 - persönlich in den Pools oder
 - über die Mailingliste
`progwerkstatt@informatik.uni-ulm.de`