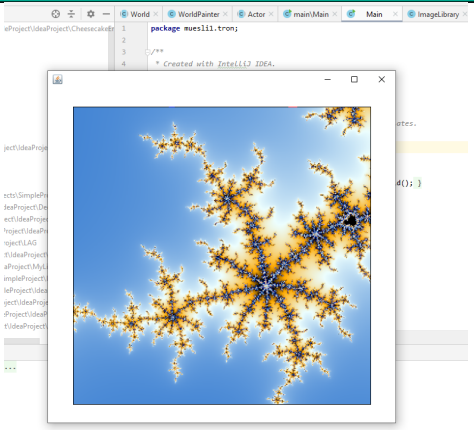
FOP Recap #8

Exceptions



Das steht heute auf dem Plan

TECHNISCHE UNIVERSITÄT DARMSTADT

- Switch
  - Statement
  - Expression
- Primitiven Wert in String umwandeln
- `Exceptions`
- `assert`
- JUnit-Tests
- Boxing

Switch
Statement

TECHNISCHE UNIVERSITÄT DARMSTADT

- Bessere Alternative zu gigantischen `if-else`-Konstrukten
- Funktionieren mit
  - Ganzzahligen Zahlen: `byte`, `short`, `char`, `int`
  - `Strings`
  - Enums

Switch

Statement — Simuliert mit `if-else`

TECHNISCHE UNIVERSITÄT DARMSTADT

```
1  int j = 5;
```

```
1  if(j == 0) {
2      // ....
3  }
4  else if(j == 1) {
5      // ....
6  }
7  else if(j == 2) {
8      // ....
9  }
10 else ....
```

Switch

Statement – Mit break

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
switch(j) {
    case 0:
        System.out.println("ZERO!");
        break;
    case 1:
        System.out.println("ONE!");
        break;

    ....

    default:
        System.out.println("DEFAULT!");
}
```

- j = 0 → ZERO!
- j = 1 → ONE!
- j = -1 → DEFAULT!

Switch

Statement — Ohne break

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
switch(j) {
    case 0:
        System.out.println("ZERO!");
        // break;
    case 1:
        System.out.println("ONE!");
        // break;

    ....

    default:
        System.out.println("DEFAULT!");
}
```

- j = 0 →
  - ZERO!
  - ONE!
  - ....
  - DEFAULT!
- j = 1 →
  - ONE!
  - ....
  - DEFAULT!
- j = -1 →
  - DEFAULT!

Switch

Statement — Enhanced Switch

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
switch(j) {
    case 0 -> {
        System.out.println("ZERO!");
        break; // nicht notwendig
    }
    case 1 -> {
        System.out.println("ONE!");
    }
    ....
    default ->
    ↪  System.out.println("DEFAULT!");
    // keine geschweiften Klammern
}
```

- `j = 0` → `ZERO!`
- `j = 1` → `ONE!`
- `j = -1` → `DEFAULT!`

Switch

Expression — Warum?

TECHNISCHE UNIVERSITÄT DARMSTADT

```
String result;

switch(j) {
    case 0:
        result = "YES";
        break;
    case 1:
        result = "OK";
        break;
    default:
        result = "MAYBE";
}
```

Switch

Expression — Mit ->

TECHNISCHE UNIVERSITÄT DARMSTADT

```
1  String result;
2
3  result = switch(j) {
4      case 0 -> "YES";
5      case 1 -> "OK";
6      default -> "MAYBE";
7  };
```

Switch

Expression — Mit `yield`

TECHNISCHE UNIVERSITÄT DARMSTADT

```
1  String result;
2
3  result = switch(j) {
4      case 0 -> {
5          yield "YES";
6      }
7      case 1 -> {
8          // ....
9          yield "OK";
10     }
11     default -> {
12         yield "MAYBE";
13     }
14 };
```

Switch

Expression — Mit `yield`

TECHNISCHE UNIVERSITÄT DARMSTADT

```
String result;

result = switch(j) {
    case 0:
        // ....
        yield "YES";
    case 1:
        // ....
        yield "OK";
    default:
        // ....
        yield "MAYBE";
};
```

Switch

Expression — Achtung: Nicht -> und : vermischen

TECHNISCHE UNIVERSITÄT DARMSTADT

```
String result;

result = switch(j) {
    case 0 -> {
        yield "YES";
    }
    default: // Error: Different case kinds used in the switch
        yield "MAYBE";
};
```

Switch

Expression — Achtung: break vergessen

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
String result = switch(new Random().nextInt(3)) {
    case 0:
        result = "YES";
    case 1:
        result = "OK";
    default:
        result = "MAYBE";
}
System.out.println(result);
```

```
$ "MAYBE"
```

- ist dann sinnvoll, wenn man den gleichen Code für mehrere Fälle ausführen möchte.
- Bei -> Syntax kann man auch mehrere Fälle angeben, diese werden dann mit , getrennt.

Das steht heute auf dem Plan

TECHNISCHE UNIVERSITÄT DARMSTADT

Switch

Primitiven Wert in String umwandeln

`Exceptions`

`assert`

JUnit-Tests

Boxing

Primitiven Wert in String umwandeln

TECHNISCHE UNIVERSITÄT DARMSTADT

```
1  int value = 0;
2  String stringValue = value; // ERROR!
```

Primitiven Wert in String umwandeln

TECHNISCHE UNIVERSITÄT DARMSTADT

```
1  int value = 0;
2  String stringValue = "" + value; // OK
```

Primitiven Wert in String umwandeln

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
int value = 0;
String stringValue = String.valueOf(value); // Super
```

Das steht heute auf dem Plan

TECHNISCHE UNIVERSITÄT DARMSTADT

- Switch
- Primitiven Wert in String umwandeln
- `Exceptions`
  - Beispiele
  - Catch-Blöcke
  - Weiterreichen
  - `RuntimeException`
  - Javadoc
  - Typische Fehler
  - Vererbung
- `assert`
- JUnit-Tests
- Boxing

Exceptions

TECHNISCHE UNIVERSITÄT DARMSTADT

- Grobe Unterscheidung zwischen
  - Fehlern
  - Laufzeitfehlern
- In Java: Viele Fehler bereits beim Kompilieren erkennbar
- Laufzeitfehler sind:
  - Fehler die erst während der Laufzeit auftreten
  - Fehler, die situationsbedingt auftreten können
- Bereits bekannte Laufzeitfehler:
  - `NullPointerException`
  - `ArrayIndexOutOfBoundsException`
  - `ClassCastException`
  - `IOException`
  - ....
- Erben alle direkt oder indirekt von `Exception`

Exceptions
Beispiele

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
public int buyCoolHat(int amountOfMoney) throws Exception {
    if(amountOfMoney < 50) {
        throw new Exception("Insufficient funds!");
    }

    System.out.println("Good choice.");
    return amountOfMoney - 50;
}
```

Exceptions
Beispiele

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
public int buyCoolHat(int amountOfMoney) throws Exception {
    // ....
}
```

```java
int myMoney = 70;

try {
    myMoney = buyCoolHat(myMoney);
}
catch(Exception e) {
    e.printStackTrace();
}
```

Exceptions
Beispiele

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
public int buyCoolHat(int amountOfMoney) throws
    NullPointerException {
    throw new NullPointerException("Not available!");
}
```

```java
int myMoney = 70;

try {
    myMoney = buyCoolHat(myMoney);
}
catch(NullPointerException e) {
    e.printStackTrace();
}
```

Exceptions
Beispiele

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
public int buyCoolHat(int amountOfMoney) throws
    InsufficientFundsException {
    // ....
}
```

```java
int myMoney = 70;

try {
    myMoney = buyCoolHat(myMoney);
}
catch(InsufficientFundsException e) {
    e.printStackTrace();
}
```

Exceptions
Beispiele

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
public int buyCoolHat(int amountOfMoney) throws
    InsufficientFundsException {
    if(amountOfMoney < 50) {
        throw new InsufficientFundsException(50 - amountOfMoney);
    }
    // ....
}
```

```java
public class InsufficientFundsException extends Exception {
    public InsufficientFundsException(int missingMoney) {
        super("Insufficient funds! Needed " + missingMoney + "
            more money.");
    }
}
```

Exceptions

Catch-Blöcke

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
public void test() throws NullPointerException,
    ClassCastException {
    // ....
}
```

```java
try {
    test();
}
catch(NullPointerException e) {
    e.printStackTrace();
}
catch(ClassCastException e) {
    e.printStackTrace();
}
```

Exceptions

Catch-Blöcke

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
public void test() throws NullPointerException,
    ClassCastException {
    // ....
}
```

```java
try {
    test();
}
catch(NullPointerException | ClassCastException e) {
    e.printStackTrace();
}
```

Exceptions
Catch-Blöcke

TECHNISCHE UNIVERSITÄT DARMSTADT

```
1  public void test() throws NullPointerException,
   ↪  ClassCastException {
2      // ....
3  }
```

```
1  try {
2      test();
3  }
4  catch(Exception e) {
5      e.printStackTrace();
6  }
```

Exceptions

Catch-Blöcke

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
public void test() throws NullPointerException,
    ClassCastException {
    // ....
}
```

```java
try {
    test();
}
catch(Exception e) {
    e.printStackTrace();
}
catch(NullPointerException e) {
    System.out.println("Null!"); // Fehler!
}
```

Exceptions

Catch-Blöcke

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
public void test() throws NullPointerException,
    ClassCastException {
    // ....
}
```

```java
try {
    test();
}
catch(NullPointerException e) {
    System.out.println("Null!");
}
catch(Exception e) {
    e.printStackTrace();
}
```

Exceptions

Catch-Blöcke

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
public void test() throws Exception {
    throw new NullPointerException("Message");
}
```

```java
try {
    test();
}
catch(NullPointerException | ClassCastException e) {
    e.printStackTrace();
}
catch(Exception e) {
    System.out.println("You need this :)");
}
```

Exceptions
Weiterreichen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Weiterreichen/Weiterleiten von Exceptions ist (unter anderem) dann sinnvoll, wenn
  - der Fehler nicht an dieser Stelle gelöst werden kann
  - es eine höhere zentrale Stelle zum Sammeln von Fehlern gibt

Exceptions
Weiterreichen

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
public void test() throws NullPointerException,
    ClassCastException {
    throw new NullPointerException("Message");
}
```

```java
public void run() throws ClassCastException {
    try {
        test();
    }
    catch(NullPointerException e) { }
    catch(ClassCastException e) {
        throw e;
    }
}
```

Exceptions

Weiterreichen

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
public void test() throws NullPointerException,
    ClassCastException {
    throw new NullPointerException("Message");
}
```

```java
public void run() throws ClassCastException {
    // Kein try und catch für ClassCastException
    try {
        test();
    }
    catch(NullPointerException e) {
        e.printStackTrace();
    }
}
```

Exceptions
RuntimeException

TECHNISCHE UNIVERSITÄT DARMSTADT

- ACHTUNG! Ausnahme!
- Alle `Exceptions` die von `RuntimeException` direkt oder indirekt erben
  - Müssen nicht mit `throws` deklariert werden
  - Müssen nicht mit einem try-and-catch Block abgeprüft werden
- Hierzu zählen zum Beispiel
  - NullPointerException
  - ArrayIndexOutOfBoundsException
  - ClassCastException
  - ....

Exceptions

Javadoc

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
/**
 * Buys a cool hat
 * @param amountOfMoney The amount of money available
 * @return Amount of money left after transaction
 * @throws InsufficientFundsException If there is not enough money
 ↪  available
 */
public int buyCoolHat(int amountOfMoney) throws
 ↪  InsufficientFundsException {
    if(amountOfMoney < 50) {
        throw new InsufficientFundsException(50 - amountOfMoney);
    }
    return amountOfMoney - 50;
}
```

Exceptions

Typische Fehler

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
public class X {
    public static int calculateTheAnswer() throws
    ↪ IllegalStateException {
        if(Y.complexCondition() == false) {
            throw new IllegalStateException("Nobody knows why
            ↪ this failed.");
        }
        return 42;
    }
}
```

Exceptions
Typische Fehler

TECHNISCHE UNIVERSITÄT DARMSTADT

```
1  public static int tryToCall() {
2      // ....
3  }
```

Exceptions

Typische Fehler

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
public static int tryToCall() {
    try {
        int returnValue = X.calculateTheAnswer();
    }
    catch(IllegalStateException e) {
        throw new RuntimeException("Look at this: " +
        ↪  e.getMessage());
    }
    return returnValue; // ERROR!
}
```

Exceptions

Typische Fehler

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
public static int tryToCall() {
    int returnValue;
    try {
        returnValue = X.calculateTheAnswer();
    }
    catch(IllegalStateException e) {
        throw new RuntimeException("Look at this: " +
        ↪  e.getMessage());
    }
    return returnValue;
}
```

Exceptions

Typische Fehler

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
public static int tryToCall() {
    try {
        return X.calculateTheAnswer();
    }
    catch(IllegalStateException e) {
        throw new RuntimeException("Look at this: " +
        ↪  e.getMessage());
    }
}
```

Exceptions
Vererbung

TECHNISCHE UNIVERSITÄT DARMSTADT

- ExceptionA erbt direkt oder indirekt von Exception
- ExceptionB erbt direkt oder indirekt von ExceptionA

```
1  public class A {
2      public void myMethod() throws ExceptionA {
3          // ...
4      }
5  }
```

Exceptions
Vererbung

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
public class B extends A {
    @Override
    public void myMethod() throws ExceptionA {
        // Gleicher Exception-Typ erlaubt
    }
}
```

```java
public class C extends A {
    @Override
    public void myMethod() throws ExceptionB {
        // "Präzisierung" erlaubt
    }
}
```

Das steht heute auf dem Plan

TECHNISCHE UNIVERSITÄT DARMSTADT

Switch
Primitiven Wert in String umwandeln
`Exceptions`
`assert`
JUnit-Tests
Boxing

assert

TECHNISCHE UNIVERSITÄT DARMSTADT

- Einfache und kurze Möglichkeit um Fehler zu werfen
- Wirft immer eine `AssertionError`, der von `Error` abgeleitet ist
- `Error` erbt weder von `Exception` noch `RuntimeException`

assert

TECHNISCHE UNIVERSITÄT DARMSTADT

- Einfache und kurze Möglichkeit um Fehler zu werfen
- Wirft immer eine `AssertionError`, der von `Error` abgeleitet ist
- `Error` erbt weder von `Exception` noch `RuntimeException`

- Ist im Normalfall deaktiviert!
- Muss mittels `-enableassertions` bzw `-ea` in den VM-Optionen aktiviert werden

assert

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
public static void dummyTest(int b) {
    assert b >= 0: "Argument must be positive!";
}
```

```java
public static int sum(int a, int b) {
    assert ((long)a + (long)b) >= Integer.MIN_VALUE &&
           ((long)a + (long)b) <= Integer.MAX_VALUE: "Overflow
            ↪ detected!";
    return a + b;
}
```

Das steht heute auf dem Plan

TECHNISCHE UNIVERSITÄT DARMSTADT

Switch

Primitiven Wert in String umwandeln

`Exceptions`

`assert`

JUnit-Tests

- Einfaches Beispiel
- `assertDoesNotThrow`
- `assertThrowsExactly`

Boxing

JUnit-Tests

TECHNISCHE UNIVERSITÄT DARMSTADT

- Framework um einfache aber auch sehr komplexe Tests für Java-Programme zu schreiben
- Library muss erst eingebunden werden
- Werden auch in Java geschrieben

JUnit-Tests

Einfaches Beispiel

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class StudentTests {
    ....
}
```

JUnit-Tests

Einfaches Beispiel

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
@Test
public void myTest() {
    Tree tree = new LemonTree();
    Apple unexpectedFruit = new Apple(5);

    assertTrue(tree.hasFruit());
    assertTrue(tree.getFruit() instanceof Lemon);
    assertNotEquals(unexpectedFruit, tree.getFruit());

    assertEquals(2, tree.getFruitAmount());
}
```

JUnit-Tests

assertDoesNotThrow

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
@Test
public void myTest() {
    HatFactory factory = new HatFactory();


    int moneyLeft = assertDoesNotThrow(
        () -> factory.buyCoolHat(52)
    );
    assertEquals(moneyLeft, 2, "Incorrect money left!");
}
```

JUnit-Tests

assertThrowsExactly

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
@Test
public void myTest() {
    HatFactory factory = new HatFactory();


    Throwable thrownException = assertThrowsExactly(
        InsufficientFundsException.class,
        () -> factory.buyCoolHat(49)
    );
    String message = thrownException.getMessage();
    // assertEquals....
}
```

Das steht heute auf dem Plan

TECHNISCHE UNIVERSITÄT DARMSTADT

Switch
Primitiven Wert in String umwandeln
`Exceptions`
`assert`
JUnit-Tests
Boxing

Boxing

TECHNISCHE UNIVERSITÄT DARMSTADT

- Für jeden primitiven Datentypen gibt es eine Wrapper-Klasse
- Später bei Generics sehr wichtig!

Primitiver Datentyp	Wrapper-Klasse
`boolean`	`java.lang.Boolean`
`byte`	`java.lang.Byte`
`short`	`java.lang.Short`
`int`	`java.lang.Integer`
`double`	`java.lang.Double`
....	....

Boxing

TECHNISCHE UNIVERSITÄT DARMSTADT

- Konvertierung in Wrapper-Klasse: Boxing
- Konvertierung zu primitiven Wert: Unboxing
- Automatisch: Auto -> Auto-Boxing und Auto-Unboxing

```
1  Integer a = ....;
2  int aPrimitive = a; // Auto-Unboxing
3
4  double bPrimitive = ....;
5  Double b = bPrimitive; // Auto-Boxing
```

Boxing

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
Integer a = Integer.valueOf(25); // Boxing
int aPrimitive = a.intValue(); // Unboxing

double bPrimitive = 5;
Double b = Double.valueOf(bPrimitive); // Boxing
```

Boxing

TECHNISCHE UNIVERSITÄT DARMSTADT

- ACHTUNG!
- Keine Widening oder Narrowing Casts bei Wrapper-Klassen
- Weder explizit noch implizit

```
1  Integer myInteger = 5; // Auto-Boxing
2  Long myLong = myInteger; // ERROR!
3  Long myOtherLong = (Long) myInteger; // ERROR!
4  Byte smallByte = myInteger; // ERROR!
5  Byte smallOtherByte = (Byte) myInteger; // ERROR!
```

Live-Coding!