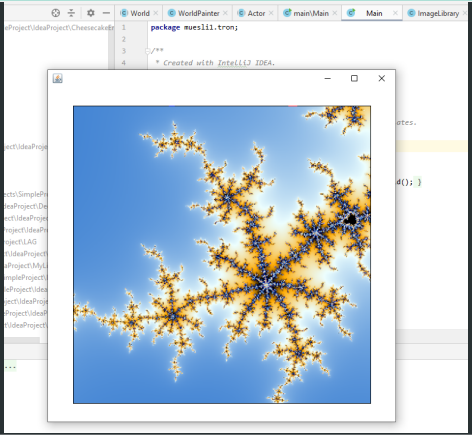
FOP Recap #9

Exceptions



Willkommen zurück!

Das steht heute auf dem Plan

TECHNISCHE UNIVERSITÄT DARMSTADT

Primitiven Wert in String umwandeln

Mehrdimensionale Arrays

`Exceptions`

`assert`

JUnit-Tests

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

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

Primitiven Wert in String umwandeln

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
int value = 0;
String stringValue = Integer.toString(value); // Exotisch
```

Das steht heute auf dem Plan

TECHNISCHE UNIVERSITÄT DARMSTADT

Primitiven Wert in String umwandeln

Mehrdimensionale Arrays
- 2D Initialisierung
- 2D Zugriff auf Elemente
- 2D Beispiel

`Exceptions`

`assert`

JUnit-Tests

Mehrdimensionale Arrays

TECHNISCHE UNIVERSITÄT DARMSTADT

- Haben als Komponentypen wieder Arrays
- Typischerweise 2D oder 3D Arrays
- Zugriff über mehrere Indices
- Wichtig: Subarrays müssen nicht die gleiche Länge haben
- Zusätzliche Kniffe fürs Erstellen

Mehrdimensionale Arrays

2D Initialisierung

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
int[][] bigArray = new int[2][5];
```

```java
int[][] bigArray = new int[2][];
bigArray[0] = new int[5];
bigArray[1] = new int[5];
```

```java
int[][] bigArray = new int[2][];
bigArray[0] = new int[] {0, 0, 0, 0, 0};
bigArray[1] = new int[] {0, 0, 0, 0, 0};
```

```java
int[][] bigArray = new int[][] {{0, 0, 0, 0, 0}, {0, 0, 0, 0, 0}};
```

Mehrdimensionale Arrays

2D Zugriff auf Elemente

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
int[][] bigArray = new int[2][5];
```

```java
int[] smallArray = bigArray[0];
smallArray[2] = 3;
```

```java
bigArray[0][2] = 3;
```

Mehrdimensionale Arrays

2D Zugriff auf Elemente

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
int[][] bigArray = new int[2][5];
```

```java
int[] smallArray = bigArray[0];
int value = smallArray[2];
```

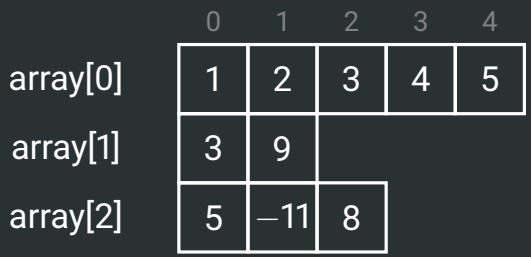
```java
int value = bigArray[0][2];
```

Mehrdimensionale Arrays

2D Beispiel

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
int[][] array = new int[3][];
array[0] = new int[] {1, 2, 3, 4, 5};
array[1] = new int[] {3, 9};
array[2] = new int[] {5, -11, 8};
```

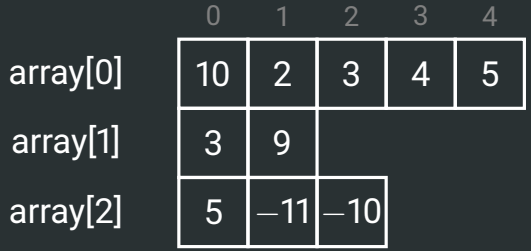


Mehrdimensionale Arrays

2D Beispiel

TECHNISCHE UNIVERSITÄT DARMSTADT

```
1  int[][] array = ....;
2  array[0][0] = 10;
3  array[2][2] = -10;
```



Mehrdimensionale Arrays

2D Beispiel

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
int[][] array = new int[3][];
array[1] = new int[] {3, 9, 10, 25};
```

Das steht heute auf dem Plan

TECHNISCHE UNIVERSITÄT DARMSTADT

Primitiven Wert in String umwandeln

Mehrdimensionale Arrays

`Exceptions`
- Einfaches Beispiel
- Besseres Beispiel
- Mehrere `Exceptions`
- Falsche Reihenfolge der Catch-Blöcke
- Korrekte Reihenfolge der Catch-Blöcke
- Mehrere `Exceptions`
- Weiterreichen
- Ausnahme `RuntimeException`

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

Einfaches Beispiel

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

Einfaches Beispiel

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

Einfaches Beispiel

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

Besseres Beispiel

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

Besseres Beispiel

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

Mehrere Exceptions

TECHNISCHE UNIVERSITÄT DARMSTADT

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

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

Exceptions

Mehrere Exceptions

TECHNISCHE UNIVERSITÄT DARMSTADT

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

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

Exceptions

Mehrere Exceptions

TECHNISCHE UNIVERSITÄT DARMSTADT

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

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

Exceptions

Falsche Reihenfolge der 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  }
7  catch(NullPointerException e) {
8      System.out.println("Null!"); // !!!
9  }
```

Exceptions

Korrekte Reihenfolge der 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

Mehrere Exceptions

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

Ausnahme `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

Primitiven Wert in String umwandeln

Mehrdimensionale Arrays

`Exceptions`

`assert`

JUnit-Tests

`assert`

- 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` 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;
}
```

Kontroverse: Exceptions in Java

TECHNISCHE UNIVERSITÄT DARMSTADT

Das steht heute auf dem Plan

TECHNISCHE UNIVERSITÄT DARMSTADT

Primitiven Wert in String umwandeln

Mehrdimensionale Arrays

`Exceptions`

`assert`

JUnit-Tests
- Einfaches Beispiel
- `assertDoesNotThrow`
- `assertThrowsExactly`

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

```
@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....
}
```

Live-Coding!