

Teil

Objektorientierte Programmierung

Unterrichtseinheit 29

JDK-Klassenbibliothek

Dr. Dietrich Boles

- JDK-Klassenbibliothek
- `java.lang.String`
- `java.lang.Integer`
- `java.util.Random`
- Zusammenfassung

- Wird von Version zu Version erweitert
- die wichtigsten Pakete:
 - java.applet: Java-Applets (Applet, ...)
 - java.awt: graphische Oberflächen (Fenster, GUI-Komponenten, Graphik, Layout-Manager, Container, ...)
 - java.awt.datatransfer: Datentransfers zwischen Applikationen (Clipboards, ...)
 - java.awt.event: Event-Handling (Maus-Events, Tastatur-Events, ...)
 - java.awt.image: Bildverarbeitung (Farbe, Filter, ...)
 - java.beans: Java-Beans-API (Properties, Introspektion, ...)
 - java.io: Ein-/Ausgabe (Streams, Dateien, ...)
 - java.lang: Basis-Klassen (System, Object, Runtime, String, ...)
 - java.lang.reflect: Java Reflection API (Introspektion, ...)

- java.math: Mathematik (Integer-, Floating-Point-Arithmetik, ...)
- java.net: Netzwerke (Sockets, URL, HTTP, ...)
- java.rmi: Client-Server-Programmierung (RMI)
- java.security: Sicherheit
- java.sql: JDBC-Datenbankzugriff
- java.text: Internationale Programme (Texte, Datum, ...)
- java.util: Nützliche Klassen (Speicher (Vector, HashTable, BitSet, Stack, ...), Datum, Random, Scanner, ...)
- javax.swing: Swing-Klassen (fortgeschrittene GUI-Programmierung)
- javax.xml: Umgang mit XML
- org.omg.CORBA: Client-Server-Programmierung (CORBA)
- ...

Dokumentation: <http://java.sun.com/j2se/1.5.0/docs/api/>



JDK-Version

```
package java.lang;

class String {
    String();
    String(char[] zeichen);
    char charAt(int index);
    String concat(String str);
    int indexOf(char zeichen);
    int length();
    String replace(char alt, char neu);
    String substring(int vonIndex, int bisIndex);
    static String valueOf(int zahl);
    ...
}
```

```
import java.lang.String; // muss nicht

class StringTest {
    public static void main(String[] args) {
        String str1 = "Hallo dibo!";
        char ch = str1.charAt(1); // ch == 'a';
        int index = str1.indexOf('o'); // index == 4
        String str2 = str1.replace('o', 'i'); // "Halli dibi!"
        str2 = str1.substring(0, 5); // "Hallo"
        str2 = String.valueOf(4711); // "4711"
    }
}
```

```
package java.lang;

class Integer {
    Integer(int value);
    static Integer valueOf(int value);

    int intValue();
    String toString();
    ...
}
```

Sinn: Behandlung von int-Werten als Objekte (→ Wrapper-Objekte)

Analoge Klassen: **Short, Long, Float, Double, Boolean, ...**

```
class Wrapper {    // Java 1.4

    public static void main(String[] args) {
        Integer[] speicher = new Integer[3];
        // speicher[0] = 4711; // Fehler bis Java 1.4
        speicher[0] = new Integer(4711);
        speicher[1] = Integer.valueOf(46);
        speicher[2] = Integer.valueOf(33);

        int summe = 0;
        for (int i = 0; i < speicher.length; i++) {
            summe += speicher[i].intValue();
        }
        System.out.println(summe);
    }
}
```



```
class Boxing { // Java 5.0

    public static void main(String[] args) {
        Integer[] speicher = new Integer[3];
        speicher[0] = 4711;
        // Autoboxing (Compiler-Rewriting)
        speicher[1] = 46;
        speicher[2] = 33;

        int summe = 0;
        for (int i = 0; i < speicher.length; i++) {
            summe += speicher[i];
            // Unboxing (Compiler-Rewriting)
        }
        System.out.println(summe);
    } }
```

```
package java.util;

class Random {
    Random(long seed); // Pseudo-Zufall
    Random();           // seed == aktuelle Zeit
    void setSeed(long seed);
    float nextFloat(); // [0..1]
    int nextInt();      // [minint.. maxint]
    ...
}
```

```
import java.util.Random;

class Wuerfel {
    Random zufall;

    Wuerfel() { this.zufall = new Random(); }

    int wuerfeln() {
        int wert = this.zufall.nextInt();
        if (wert < 0) wert = -wert;
        return (wert%6) + 1;
    }

    public static void main(String[] args) {
        Wuerfel wuerfel = new Wuerfel();
        for (int i=0; i<5; i++)
            IO.println(wuerfel.wuerfeln());
    } }
```

- JDK-Klassenbibliothek: standardisierte Sammlung von zahlreichen nützlichen Klassen für vielfältige Anwendungsgebiete