Programmierkurs Java

UE 16 – JDK Klassenbibliothek

Dr.-Ing. Dietrich Boles

Gliederung

- Übersicht
- java.lang.String
- java.lang.Integer
- Boxing
- java.util.ArrayList
- Zusammenfassung

Übersicht (1)

- 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, ...)

Übersicht (2)

- 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: https://docs.oracle.com/en/java/javase/index.html

java.lang.String (1)

```
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);
    ...
}
```

java.lang.String (2)

```
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"
  }
}
```

java.lang.Integer

```
package java.lang;

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

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

int intWert1 = 47;
Integer intObj = Integer.valueOf(intWert1); ...
int intWert2 = intObj.intValue();
```

Sinn: Behandlung von int-Werten als Objekte (→ Wrapper-Objekte)
Analoge Klassen: **`Short, Long, Float, Double, Boolean, ...`**

Boxing (1)

```
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);
  }
}
```

Boxing (2)

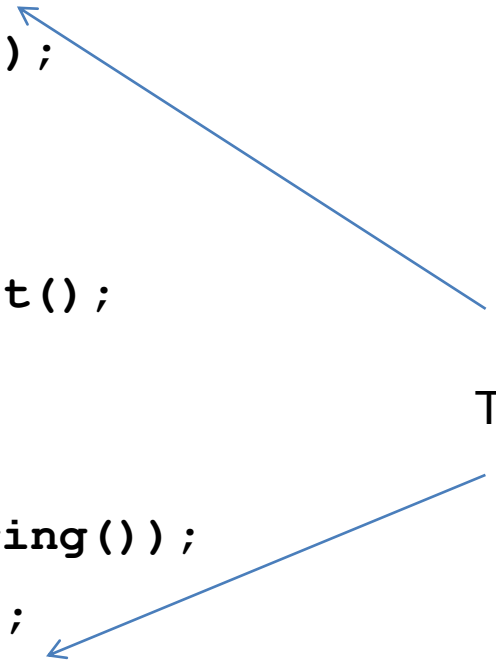
```
class Boxing { // Java 5.0

  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);
} }
```

java.util.ArrayList (1)

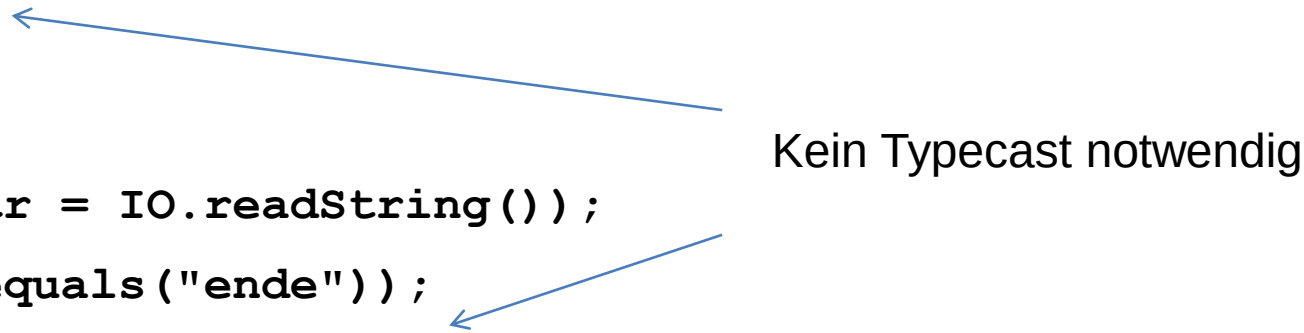
```
package java.util;
class ArrayList { // in der Version Java 1.4 (ohne Generics)
    ArrayList();
    void add(Object o); // Polymorphie
    Object get(int index);
    boolean contains(Object o);
    int size(); ...
}
ArrayList list = new ArrayList();
String str;
do {
    list.add(str = IO.readString());
} while (!str.equals("ende"));
String ersterString = (String) list.get(0);
```



Typecast notwendig

java.util.ArrayList (2)

```
package java.util;
class ArrayList<T> {// mit Generics, T = bel. Referenztyp
    ArrayList();
    void add(T o);
    T get(int index);
    boolean contains(T o);
    int size(); ...
}
ArrayList<String> list = new ArrayList<String>();
String str;
do {
    list.add(str = IO.readString());
} while (!str.equals("ende"));
String ersterString = list.get(0);
```



java.util.ArrayList (3)

```
ArrayList<Integer> list = new ArrayList<Integer>();


for (int i = IO.readInt(); i >= 0; i--) {
    list.add(i * i);
}
int value = list.get(0);


for (int value : list) {
    System.out.println(value);
}
```

Boxing (int ←→ Integer)

Nutzung der for-each-Schleife

Zusammenfassung

- JDK-Klassenbibliothek:

  standardisierte Sammlung von zahlreichen nützlichen Klassen für vielfältige Anwendungsgebiete

- Boxing:

  Automatische Umwandlung von Standard- in Referenzdatentypen und umgekehrt

- java.util.ArrayList:

  Klasse für flexibel große Container-Objekte (Alternative zu Arrays)