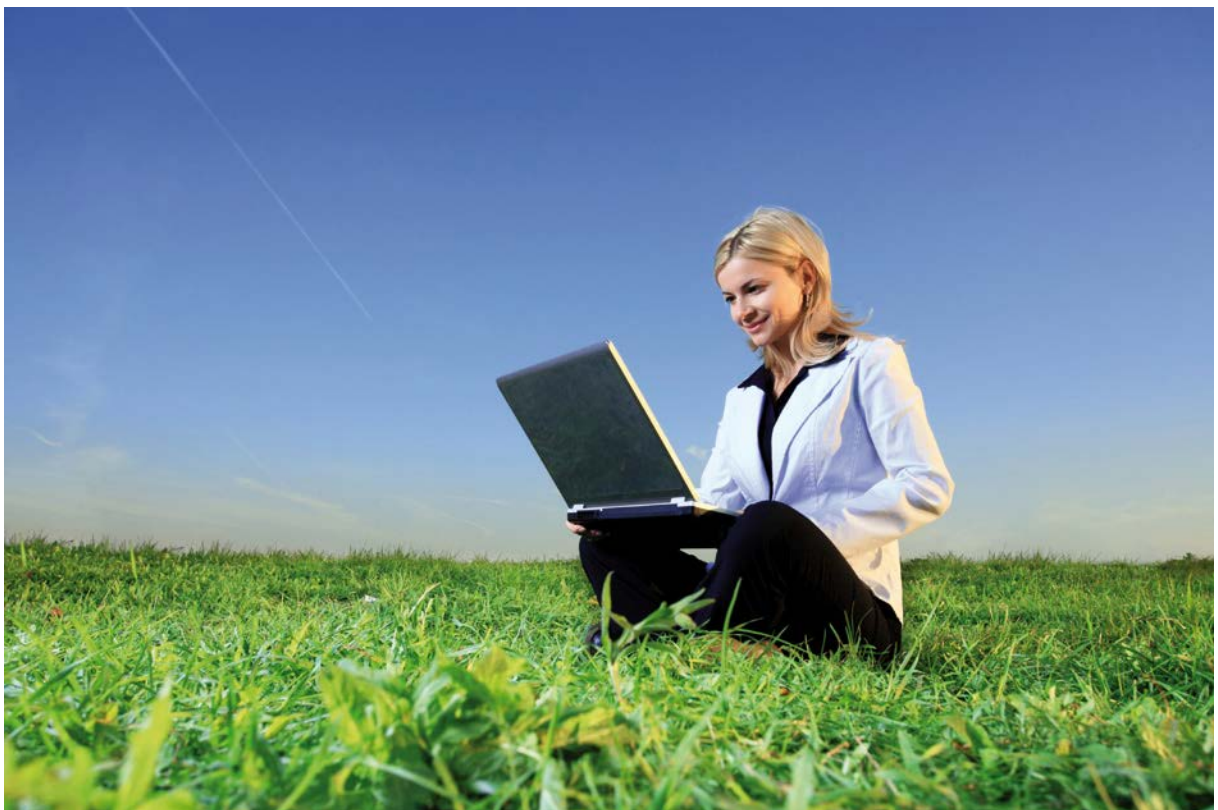


# ***Java Programmierung Aufbau***

***Seminarunterlage***

***Version: 11.03***



Dieses Dokument wird durch die ORDIX AG veröffentlicht.

Copyright ORDIX AG. Alle Rechte vorbehalten.

Alle Produkt- und Dienstleistungs-Bezeichnungen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Firmen und beziehen sich auf Eintragungen in den USA oder USA-Warenzeichen.

Weitere Logos und Produkt- oder Handelsnamen sind eingetragene Warenzeichen oder Warenzeichen der jeweiligen Unternehmen.

Kein Teil dieser Dokumentation darf ohne vorherige schriftliche Genehmigung der ORDIX AG weitergegeben oder benutzt werden.

## Adressen der ORDIX AG

Die ORDIX AG besitzt folgende Geschäftsstellen

ORDIX AG  
Westernmauer 12-16  
D-33098 Paderborn  
Tel.: (+49) 0 52 51 / 10 63 - 0  
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG  
Kreuzberger Ring 13  
D-65205 Wiesbaden  
Tel.: (+49) 06 11 / 7 78 40 – 00  
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG  
An der alten Ziegelei 5  
D-48157 Münster  
Tel.: (+49) 02 51 / 9 24 35 – 00  
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG  
Wikingerstraße 18-20  
D-51107 Köln  
Tel.: (+49) 02 21 / 8 70 61 – 0  
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG  
Marlene-Dietrich-Str. 5  
D-89231 Neu-Ulm  
Tel.: (+49) 07 31 / 9 85 88 – 550  
Fax.: (+49) 01 80 / 1 67 34 90

Sie können die ORDIX AG von der ganzen Welt aus durch folgende Internet Adresse kontaktieren:  
<http://www.ordix.de>

Sie können uns weiterhin über die Email-Adressen [training@ordix.de](mailto:training@ordix.de) oder [info@ordix.de](mailto:info@ordix.de) kontaktieren

## Inhaltsverzeichnis

|          |                                                         |           |
|----------|---------------------------------------------------------|-----------|
| <b>1</b> | <b>Grundlagen .....</b>                                 | <b>7</b>  |
| 1.1      | Was ist Java? .....                                     | 8         |
| 1.2      | Historie .....                                          | 9         |
| 1.3      | Java-Design Kriterien .....                             | 10        |
| 1.3.1    | Einfach und Objektorientiert .....                      | 11        |
| 1.3.2    | Verteilt, Interpretiert, Robust .....                   | 12        |
| 1.3.3    | Sicher, Architekturneutral, Portabel .....              | 13        |
| 1.3.4    | Hochleistungsfähig und Multithreaded .....              | 14        |
| 1.3.5    | Dynamisch .....                                         | 15        |
| 1.4      | Java Virtual Machine (JVM) .....                        | 16        |
| 1.5      | Was macht die JVM? .....                                | 17        |
| 1.6      | *.java und *.class-Datei .....                          | 18        |
| 1.7      | Java-Anwendungen .....                                  | 19        |
| 1.8      | Einfache Java-Anwendung .....                           | 20        |
| 1.9      | Kompilieren und Starten .....                           | 21        |
| 1.10     | Kommentare .....                                        | 22        |
| 1.11     | Programmierertools .....                                | 23        |
| <b>2</b> | <b>Collection Framework .....</b>                       | <b>24</b> |
| 2.1      | Was ist eine Collection? .....                          | 25        |
| 2.2      | Architektur des Collection Frameworks .....             | 26        |
| 2.3      | Collection Framework Interfaces .....                   | 27        |
| 2.3.1    | Interface Collection .....                              | 29        |
| 2.3.2    | Interface List .....                                    | 30        |
| 2.3.3    | Interface Set .....                                     | 31        |
| 2.3.4    | Interface SortedSet .....                               | 32        |
| 2.3.5    | Interface Map .....                                     | 33        |
| 2.3.6    | Interface SortedMap .....                               | 34        |
| 2.4      | LinkedList .....                                        | 35        |
| 2.5      | ArrayList .....                                         | 37        |
| 2.6      | HashSet .....                                           | 39        |
| 2.7      | TreeSet .....                                           | 40        |
| 2.8      | HashMap .....                                           | 41        |
| 2.9      | TreeMap .....                                           | 43        |
| 2.10     | Interface Iterator .....                                | 45        |
| 2.11     | Interface ListIterator .....                            | 47        |
| 2.12     | Klasse Collections .....                                | 50        |
| 2.13     | Algorithmen – Sortieren .....                           | 51        |
| 2.14     | Algorithmen – Permutieren .....                         | 52        |
| 2.15     | Algorithmen - Minimum & Maximum .....                   | 53        |
| 2.16     | Algorithmen – Suchen .....                              | 54        |
| 2.17     | Algorithmen – Datenmanipulation .....                   | 55        |
| 2.18     | Synchronisierte Wrapper .....                           | 56        |
| 2.19     | Nicht modifizierbare Wrapper .....                      | 58        |
| 2.20     | Zusätzliche Funktionalität .....                        | 59        |
| 2.21     | Einbindung bestehender Collections .....                | 60        |
| 2.22     | Aufwärts – Kompatibilität .....                         | 62        |
| 2.23     | Abwärts – Kompatibilität .....                          | 64        |
| <b>3</b> | <b>Generics .....</b>                                   | <b>66</b> |
| 3.1      | Java Generics .....                                     | 67        |
| 3.2      | Collections bisher .....                                | 68        |
| 3.3      | Generische Collections .....                            | 69        |
| 3.4      | Generische Collections mit mehreren Typparametern ..... | 70        |
| 3.5      | Ober- und Unterklassen in generischen Typen .....       | 71        |
| 3.6      | weak und strong typing .....                            | 73        |
| 3.7      | Die „andere Seite“ .....                                | 74        |

|          |                                                          |            |
|----------|----------------------------------------------------------|------------|
| 3.8      | Eigene generische Typen .....                            | 76         |
| 3.9      | TreeMap ohne Bounds.....                                 | 80         |
| 3.10     | Bounds .....                                             | 84         |
| 3.11     | TreeMap mit Bounds.....                                  | 85         |
| 3.12     | Generische Methoden.....                                 | 87         |
| 3.12.1   | generische Methoden Java 1.4 .....                       | 88         |
| 3.12.2   | generische Methoden Java 1.5 .....                       | 90         |
| 3.13     | Typ-Inferenz .....                                       | 92         |
| 3.14     | Wildcard Instanziierung.....                             | 94         |
| 3.15     | Wildcard Beispiele.....                                  | 96         |
| 3.16     | Type Erasure.....                                        | 99         |
| 3.17     | Generische Typen – Grenzen .....                         | 101        |
| 3.18     | Zusammenfassung.....                                     | 104        |
| <b>4</b> | <b>Objektserialisierung.....</b>                         | <b>105</b> |
| 4.1      | Objektserialisierung.....                                | 106        |
| 4.2      | Serialisierung eines Objektes.....                       | 107        |
| 4.3      | Deserialisierung eines Objektes.....                     | 109        |
| 4.4      | Voraussetzungen zur Objektserialisierung .....           | 111        |
| 4.5      | Dateiformat der Objektserialisierung.....                | 112        |
| 4.6      | Problem der Speicherung von Objektreferenzen.....        | 115        |
| 4.7      | Prinzip der Speicherung von Objektreferenzen .....       | 117        |
| 4.8      | Sicherheit .....                                         | 118        |
| 4.9      | Erweiterung der Serialisierung und Deserialisierung..... | 120        |
| 4.10     | Interface Externalizable.....                            | 122        |
| 4.11     | Sicherheitstipps.....                                    | 123        |
| 4.12     | Versionsverwaltung.....                                  | 124        |
| 4.13     | Serialisierung zum Klonen .....                          | 126        |
| <b>5</b> | <b>Innere Klassen .....</b>                              | <b>128</b> |
| 5.1      | Was sind Innere Klassen? .....                           | 129        |
| 5.2      | „Normale“ Toplevel Klassen.....                          | 130        |
| 5.3      | Beispiel: Innere Klassen.....                            | 131        |
| 5.4      | Typisierung von Inneren Klassen.....                     | 132        |
| 5.5      | Statische innere Klassen.....                            | 133        |
| 5.6      | Nicht-statische innere Klassen.....                      | 135        |
| 5.7      | Member Klassen .....                                     | 136        |
| 5.8      | Beispiel: Member Klassen.....                            | 138        |
| 5.9      | Lokale Klassen .....                                     | 139        |
| 5.10     | Beispiel: Lokale Klassen .....                           | 141        |
| 5.11     | Anonyme Klassen .....                                    | 142        |
| 5.12     | Beispiel: Anonyme Klassen.....                           | 144        |
| 5.13     | Anwendung: Implementation Hiding .....                   | 145        |
| 5.14     | Anwendung: Adapterklassen .....                          | 146        |
| 5.15     | Zusammenfassung.....                                     | 147        |
| <b>6</b> | <b>Typesafe Enums .....</b>                              | <b>148</b> |
| 6.1      | Typesafe Enums .....                                     | 149        |
| 6.2      | Aufzählungen bisher .....                                | 150        |
| 6.3      | Probleme mit int-Enumerations.....                       | 151        |
| 6.4      | Idiom für typsichere Aufzählungen.....                   | 153        |
| 6.5      | Typesafe Enums .....                                     | 157        |
| 6.6      | Deklaration von enum-Klassen .....                       | 159        |
| 6.7      | Eigenschaften von enum-Klassen.....                      | 161        |
| 6.8      | enum Beispiel.....                                       | 164        |
| 6.9      | Methoden in enum-Klassen .....                           | 168        |
| 6.9.1    | switch-Statement.....                                    | 170        |
| 6.9.2    | abstrakte Methoden.....                                  | 172        |
| 6.10     | Zwei neue Klassen für enum.....                          | 173        |

|                                                              |            |
|--------------------------------------------------------------|------------|
| 6.10.1 EnumSet.....                                          | 174        |
| 6.10.2 EnumMap.....                                          | 178        |
| <b>7 Reflection API.....</b>                                 | <b>181</b> |
| 7.1 Reflection API.....                                      | 182        |
| 7.2 Klassen der Reflection API.....                          | 183        |
| 7.3 Klassen erkunden.....                                    | 186        |
| 7.4 Klasse ermitteln.....                                    | 187        |
| 7.5 Klassennamen ermitteln.....                              | 189        |
| 7.6 Klassenmodifizierer ermitteln.....                       | 190        |
| 7.7 Superklasse ermitteln.....                               | 192        |
| 7.8 Interface einer Klasse ermitteln.....                    | 193        |
| 7.9 Interfaces-Eigenschaft abprüfen.....                     | 194        |
| 7.10 Klassen- bzw. Interfaceattribute ermitteln.....         | 195        |
| 7.11 Konstruktoren ermitteln.....                            | 197        |
| 7.12 Methoden samt Signatur ermitteln.....                   | 198        |
| 7.13 Objekte manipulieren.....                               | 199        |
| 7.14 Objekte über den Default-Konstruktor erzeugen.....      | 200        |
| 7.15 Objekte über parametrisierten Konstruktor erzeugen..... | 201        |
| 7.16 Attributwerte ermitteln.....                            | 203        |
| 7.17 Attributwerte setzen.....                               | 205        |
| 7.18 Methoden aufrufen.....                                  | 207        |
| 7.19 Arbeiten mit Arrays.....                                | 209        |
| 7.20 Array-Eigenschaft abprüfen.....                         | 210        |
| <b>8 Java Database Connectivity (JDBC).....</b>              | <b>212</b> |
| 8.1 Zugriff auf Datenbanken in der Nicht-Java-Welt.....      | 213        |
| 8.2 Behandlung verschiedener Datenbanken.....                | 214        |
| 8.3 Open Database Connectivity (ODBC).....                   | 215        |
| 8.4 Java Database Connectivity (JDBC).....                   | 216        |
| 8.5 Eigenschaften des JDBC.....                              | 217        |
| 8.6 Beziehung zwischen JDBC-Klassen.....                     | 218        |
| 8.7 DB-Verbindung aufbauen.....                              | 220        |
| 8.8 SQL-Anweisungen absetzen.....                            | 222        |
| 8.9 Interface Statement.....                                 | 223        |
| 8.10 Beispiel: SQL-Anweisung absetzen.....                   | 224        |
| 8.11 Auslesen einer Ergebnismenge.....                       | 226        |
| 8.12 Interface ResultSet.....                                | 227        |
| 8.13 Beispiel: Auslesen der Ergebnismenge.....               | 228        |
| 8.14 Zuordnung JDBC Typ zu Java Typ.....                     | 230        |
| 8.15 Schließen einer DB-Verbindung.....                      | 231        |
| 8.16 Vorbereitete Anweisungen.....                           | 232        |
| 8.17 JDBC und Transaktionen.....                             | 234        |
| 8.18 Transaktions-Modi.....                                  | 235        |
| 8.19 Beispiel mit Transaktionen.....                         | 237        |
| 8.20 JDBC und Metadaten.....                                 | 239        |
| 8.21 Interface DatabaseMetaData.....                         | 240        |
| 8.22 Interface ResultSetMetaData.....                        | 241        |
| 8.23 Zusammenfassung.....                                    | 243        |
| <b>9 Annotations.....</b>                                    | <b>244</b> |
| 9.1 Annotations.....                                         | 245        |
| 9.2 Was sind Annotations?.....                               | 246        |
| 9.3 Anwendungsgebiete.....                                   | 248        |
| 9.4 Annotation Beispiel: @Deprecated.....                    | 250        |
| 9.5 Verwendung von Annotations.....                          | 252        |
| 9.6 Packages für Annotations.....                            | 254        |
| 9.7 Vordefinierte Annotations.....                           | 256        |
| 9.8 Standard Annotations.....                                | 257        |

|           |                                                      |            |
|-----------|------------------------------------------------------|------------|
| 9.9       | Beispiel: @SuppressWarnings.....                     | 259        |
| 9.10      | Definition eigener Annotations-Typen.....            | 261        |
| 9.11      | Annotations mit Eigenschaften.....                   | 263        |
| 9.12      | Meta-Annotation Target .....                         | 268        |
| 9.13      | Beispiel: @Target.....                               | 272        |
| 9.14      | Meta-Annotation Retention .....                      | 273        |
| 9.15      | Beispiel: @Retention.....                            | 276        |
| 9.16      | Meta-Annotation Documented .....                     | 277        |
| 9.17      | Beispiel: @Documented.....                           | 279        |
| 9.18      | Meta-Annotation Inherited.....                       | 281        |
| 9.19      | Beispiel: @Inherited .....                           | 282        |
| 9.20      | Zugriff über Reflection API .....                    | 284        |
| 9.21      | Interface AnnotatedElement.....                      | 285        |
| 9.22      | Vorgehensweise mit Reflection.....                   | 286        |
| 9.23      | Beispiel: Zugriff über Reflection .....              | 287        |
| <b>10</b> | <b>Garbage Collection.....</b>                       | <b>290</b> |
| 10.1      | Was ist die Garbage Collection (GC)?.....            | 291        |
| 10.2      | finalize()-Methode .....                             | 292        |
| 10.3      | „Teilweise“ und „Vollständige“ GC .....              | 293        |
| 10.4      | Unterschiedlichste Parametereinstellungen.....       | 294        |
| 10.5      | Anpassung der initialen und max. Speichergröße ..... | 295        |
| 10.6      | Verfolgen der Garbage Collection Aktivität .....     | 296        |
| 10.7      | „Generational Virtual Machine“ .....                 | 297        |
| 10.8      | Anpassen der „nursery“-Speichergröße.....            | 298        |
| 10.9      | „Incremental Mode“ .....                             | 299        |