

Objektorientierte Programmierung mit Java für die IT Berufe des dualen Systems und Informationstechnische Assistenten

Geschichte, zentrale Eigenschaften und Installation von Java SE & IDE

- Editionen
 - Micro Edition (ME)
 - Standard Edition (SE)
 - Enterprise Edition (EE)
- 3DK, JRE, JVM
- Vom Quellcode zum Bytecode
- Standard-Klassenbibliothek
- Java API-Dokumentation
- IDEs und Editoren
 - NetBeans
 - Eclipse
 - IntelliJ IDEA
 - BlueJ
 - JavaEditor
- Eigenschaften und Merkmale
 - objektorientiert
 - plattformunabhängig
 - verteilbar
 - abstraktionierbar
 - automatisches Speichermanagement
 - Java Security Modell
 - Open Source
 - starke Verbreitung
 - viele Bibliotheken und Frameworks

Sprachelemente in Java

- Bezeichner
 - Literale
 - Schlüsselwörter
 - Kommentare
 - API - Dokumentation
 - Klasse
 - main - Methode
 - Anweisungen
 - Anweisungsblicke
 - Aufruf statischer Methoden
 - Modifizierer
- Ein- und Ausgaben
 - Dialogfenster
 - Konsole

Variablen und Datentypen

- Primitive Datentypen
 - ganzzahlige
 - char
 - byte
 - short
 - int
 - long
 - Wahrheitswerte **boolean**
 - Fließkommazahlen
 - double
 - float
- Die Klasse String
 - Delimitation
 - Wertzuweisung
 - Bezeichner
- Variablen
 - lokale Variablen
 - Typenpassung (Casting)
 - Typkonvertierung (Parsing)

Methoden

- Parameter, Rückgabewert und Rückgabewert
 - Wertübergabe **call-by-value**
 - einfache Ausnahmebehandlung
 - statische Methoden
 - return - Anweisung
 - Konstanten **final**
 - Klassenvariablen **static**
 - lokale Variablen und Gültigkeit
 - Rekursive Methoden
 - VarArgs

Audrucke, Operatoren

- Mathematische Ausdrücke
- Relationale Operatoren
- Logische Operatoren
- Arithmetische Operatoren
- Zuweisungen mit Operation (Kurzschreibweisen)
- Methoden der Klasse java.lang.Math
 - round()
 - ceil()
 - floor()
 - random()
 - ...

Kontrollstrukturen

- Typkontrolle
 - Typenpassung (Casting)
 - Typkonvertierung (Parsing)
- bedingte Anweisungen
 - if
 - if...else
 - if...else if...else
- Fallunterscheidung **switch**

Wiederholungsstrukturen

- while-Schleife
- do...while-Schleife
- for-Schleife
- for-each-Schleife
- Abbruchbedingungen
 - break
 - continue

String-Validierung und einfache Fehlerbehandlung

- wichtige Stringmethoden
 - NullPointerException **e = null**
 - bestimmtes Zeichen **charAt()**
 - Stringvergleich **equals()**, **equalsIgnoreCase()**, **compareTo()**
 - Zeichen ersetzen **replaceAll()**, **replace()**
 - Sandstriche ändern **toLowerCase()**, **toUpperCase()**
 - Stringteile extrahieren **substring()**
 - Stringlänge **length()**, **isEmpty()**
 - Zeichenfolgen suchen **contains()**
 - Pattern matching **matches()**
 - Leerzeichen entfernen **trim()**
 - Stringformatierung **format()**
 - String trennen nach Delimiter **trim()**
 - Repräsentation in String **static valueOf()**
- einfache Ausnahmebehandlung **try-catch-Block**

Darstellungsformen

- UML-Diagramme
 - UML-Klassendiagramm
 - UML-Objektdiagramm
 - UML-Awendungsfalldiagramm
 - UML-Sequenzdiagramm
 - UML-Aktivitätsdiagramm
- Diagrammarten
 - Nezvi-Shneiderman-Diagramm (Struktogramm)
 - Programmablaufplan (PAP) / Flussdiagramm (engl. Flowchart)

Unified Modeling Language (UML)

- Klassendiagramm
- Objektdiagramm
- Anwendungsfalldiagramm
- Sequenzdiagramm
- Aktivitätsdiagramm

Klassen und Objekte

- Zugriffsmodifizierer
 - public
 - private
 - protected
 - package private
- Datenkapselung
 - private
 - protected
 - package private
- inhalte und statische Variablen
- Packages
- komplexe Datentypen
- Konstruktoren
 - Standardkonstruktor
 - parametrisierte Konstruktoren
 - Vererbung
- Enums
- Objektidentität, Gleichheit und equals()
- Überladene Methoden
- Referenzen
- Wrapperklassen
- innere Klasse

Objektorientierte Beziehungen

- Vererbung
 - Vererbungshierarchien
 - Generalisierung
 - Spezialisierung
 - Annotation **@Override**
 - Aufruf von Methoden der Superklassen **super.**
 - Klasse Object
 - toString()
 - equals()
- Polymorphie
 - dynamische Methodenbindung
 - Casting komplexer Datentypen
- abstrakte Klassen / Methoden
 - abstrakt
 - abstrakte Methoden
- finale Klassen / Methoden
 - final class
 - final method
- Isa-Kopplung
- abstrakte Methoden
 - funktionale Interfaces
 - Lambda-Ausdrücke
 - @FunctionalInterface
- Default Methoden
- Komposition
 - Fabrik-Methode
- Assoziationen
 - Aggregation

Entwurfsmuster

- Singleton
- MVC
- Beobachter
- Strategie
- Erbauer
- Fabrik

Datenbankschnittstelle JDBC

- Konzeption
- Connection
- Statement und PreparedStatement
- ResultSet
- try-with-resources (try-with-resources)
- DAO-Pattern

Dateien und Datenströme

- Serialisierung
- URL/URLConnection
- Dateien und Verzeichnisse
- Stream-Klassen

GUI-Programmierung

- AWT
- Swing
- JavaFX

Vorgehensmodelle der Softwareentwicklung

- klassische Methoden
 - Phasenmodell
 - Wasserfallmodell
 - Spiralmodell
 - V-Model (V)
- agile Methoden
 - XP
 - SCRUM
 - Kanban
 - ...

Qualitätskriterien für Software

- Richtigkeit
- Sicherheit
- Fehlertoleranz
- Verständlichkeit
- Eignbarkeit
- Benutzbarkeit
- Bedienbarkeit
- Verhalten
- Effizienz
- Skalierbarkeit
- Profielbarkeit
- Modifizierbarkeit
- Wertbarkeit
- Anpassbarkeit
- Austauschbarkeit
- Übertragbarkeit
- Installierbarkeit

Softwaretests

- automatisiertes Testen
 - JUnit-Tests
- Blackbox-Testing
 - Bildung von Äquivalenzklassen
 - Grenzwertanalyse
 - Einzel-Caseing
 - Ursache-Wirkungs-Graph
- Whitebox-Testing
 - Anweisungsberechtigungs-test (CO - Test)
 - Zweigüberdeckungs-test (C1 - Test)
 - Flußüberdeckungs-test (C2 - Test)
 - Bedingungsüberdeckungs-test (CIA - und CIA/CIC - Test)
- Teststufen
 - Komponententest
 - Integrationstest
 - Funktions-test
 - Systemtest
 - Performance-Test
 - Stabilitätstest
 - Abnahmetest
- Testprozess
 - Testplanung
 - Testspezifikation
 - Testdurchführung
 - Testdokumentation
 - Testauswertung
 - Testwiederholung
- Ziele
 - Kosten der Vorbeugung, Prüfung
 - Fehlerfolgskosten
 - Optimaler Qualitätsgrad

Datenstrukturen / Collections

- Listen
- Maps
- Trees
- Iteratoren
- Streams
- Generics
- generisch typisierte Datenstrukturen

Entwurfsmuster

- Singleton
- MVC
- Beobachter
- Strategie
- Erbauer
- Fabrik

Datenbankschnittstelle JDBC

- Konzeption
- Connection
- Statement und PreparedStatement
- ResultSet
- try-with-resources (try-with-resources)
- DAO-Pattern

Dateien und Datenströme

- Serialisierung
- URL/URLConnection
- Dateien und Verzeichnisse
- Stream-Klassen

GUI-Programmierung

- AWT
- Swing
- JavaFX

Vorgehensmodelle der Softwareentwicklung

- klassische Methoden
 - Phasenmodell
 - Wasserfallmodell
 - Spiralmodell
 - V-Model (V)
- agile Methoden
 - XP
 - SCRUM
 - Kanban
 - ...

Qualitätskriterien für Software

- Richtigkeit
- Sicherheit
- Fehlertoleranz
- Verständlichkeit
- Eignbarkeit
- Benutzbarkeit
- Bedienbarkeit
- Verhalten
- Effizienz
- Skalierbarkeit
- Profielbarkeit
- Modifizierbarkeit
- Wertbarkeit
- Anpassbarkeit
- Austauschbarkeit
- Übertragbarkeit
- Installierbarkeit

Softwaretests

- automatisiertes Testen
 - JUnit-Tests
- Blackbox-Testing
 - Bildung von Äquivalenzklassen
 - Grenzwertanalyse
 - Einzel-Caseing
 - Ursache-Wirkungs-Graph
- Whitebox-Testing
 - Anweisungsberechtigungs-test (CO - Test)
 - Zweigüberdeckungs-test (C1 - Test)
 - Flußüberdeckungs-test (C2 - Test)
 - Bedingungsüberdeckungs-test (CIA - und CIA/CIC - Test)
- Teststufen
 - Komponententest
 - Integrationstest
 - Funktions-test
 - Systemtest
 - Performance-Test
 - Stabilitätstest
 - Abnahmetest
- Testprozess
 - Testplanung
 - Testspezifikation
 - Testdurchführung
 - Testdokumentation
 - Testauswertung
 - Testwiederholung
- Ziele
 - Kosten der Vorbeugung, Prüfung
 - Fehlerfolgskosten
 - Optimaler Qualitätsgrad

Datenstrukturen / Collections

- Listen
- Maps
- Trees
- Iteratoren
- Streams
- Generics
- generisch typisierte Datenstrukturen