

Programmierung in Java und Java EE für Mainframe Entwickler*innen

Zielgruppe

- **Anwendungsentwickler*innen**
- **COBOL Anwendungsentwickler*innen**

Version 1.1 vom August 2022

Autoren: Wolfram Greis, Prof. Dr. Philipp Brune

European Mainframe Academy GmbH
Am Kloostergarten 3
D 78337 Öhningen
Tel. +49-7735-9388668
education@mainframe-academy.de

European Mainframe Academy AG
Obergass 23
CH 8260 Stein am Rhein
Tel. +41-52-558 20 40
education@mainframe-academy.de

Inhaltsverzeichnis

1	Ziele des Ausbildungsmoduls	3
2	Randbedingungen / Voraussetzungen	3
3	Inhalt.....	4
3.1	Java Grundlagen.....	4
3.2	Java Enterprise Edition (Java EE).....	6

1 Ziele des Ausbildungsmoduls

Ziel des Moduls ist es, Teilnehmern die Programmiersprache **Java** beizubringen. Hierbei wird speziell die Zielgruppe Entwickler*innen für den IBM Mainframe angesprochen. Wir haben speziell Erfahrung in der Ausbildung von COBOL Entwickler*innen und da unsere Referenten sowohl Java als auch COBOL beherrschen, wird die Erarbeitung der Inhalte für die Teilnehmer*innen deutlich erleichtert und äusserst effizient. An allen relevanten Stellen werden die Gemeinsamkeiten und Unterschiede hervorgehoben.

2 Randbedingungen / Voraussetzungen

Kenntnisse der grundlegenden Logik der Programmierung und von Algorithmen wird vorausgesetzt.

Den Teilnehmern wird für die Dauer der Ausbildung ein Zugang auf dem z/OS System der EMA (aktuell: z/OS 2.4) zur Verfügung gestellt. Dieser Zugang steht auch nach Beendigung der Ausbildung zur Verfügung. Die EMA garantiert diesen Zugang für mindestens drei Monate nach Ausbildungsende.

Wir setzen ausserdem voraus, dass die Teilnehmer*innen diverse Werkzeuge auf ihren Workstations installieren können.

Als Begleitmaterial werden Unterlagen und Folien der EMA und von Partnerunternehmen eingesetzt. Ergänzend wird auf IBM Redbooks und weitere Literatur verwiesen.

Die Ausbildung wird nach dem bewährten Blended-Learning Konzept der EMA durchgeführt. Nähere Informationen dazu gibt es auf unserer Website:

www.mainframe-academy.de

Die Ausbildung ist in zwei Blöcke unterteilt:

- Java Grundlagen (ca. 5 Wochen)
- Java EE (Jakarta) ca. 5 Wochen

Es wird von den Teilnehmer*innen ein Arbeitsaufwand von **durchschnittlich** 10 Stunden / Woche erwartet.

3 Inhalte

3.1 Java Grundlagen (ca. 5 Wochen)

Java Grundkonzepte

- Historie höherer Programmiersprachen
- Ein erstes Java-Programm "zu Fuß" (ohne IDE)

Java Syntax und elementare Strukturen

- Variablen, Datentypen und Konstanten
- Elementare Datentypen vs. Klassen
- Java Konventionen
- Anweisungen, Ausdrücke und Blöcke
- Operatoren in Java

Objektorientierte Programmierung

- Was ist ein "Objekt"?
- Attribute (Eigenschaften, Zustand) und Methoden (Operationen)
- Kapselung
- Klassen und Objekte in Java
- Erstellung von Klassen

Syntaktische Struktur von Java-Klassen

- Syntax und Semantik
- Grammatik
- Hierarchische Schachtelung und Einrückung
- Programmierrichtlinien und Konventionen

Java Kontrollstrukturen

- Boole'sche Ausdrücke und Bedingungen
- Verzweigungen
- Switch-Anweisung
- Schleifen
- Abweisende / vorprüfende Schleife
- Annehmende / nachprüfende Schleife
- Zählschleife (for-Schleife)

Fehler und Fehlerbehandlung

- Syntaktische Fehler
- Laufzeitfehler
- Exception und Exception-Mechanismus
- Fehlerbehandlung (throw und catch)
- try-catch-Anweisung

Verwaltung und Verarbeitung von strukturierten Mengen

- Felder, Listen und Collections
- Statisches Array (Liste feste Länge)
- Dynamisches Array (Liste variable Länge)
- Assoziatives Array (HashMap)
- for-each-Schleife (Iteration über Listen)

Objekte und Klassen - Vertiefung

- Klassen als Datentypen
- Objektvariable this

Vererbung

- Spezialisierung / Generalisierung
- Substitutionsprinzip nach Barbara Liskov
- Überschreiben von Methoden
- Polymorphie (Vielgestaltigkeit)
- Kapselungsprinzip
- private/public/protected

Interfaces und Generics

- Interfaces und Realisierung
- "Verträge" zwischen Klassen
- Generische Klassen
- "Vorlagen" für konkrete Klassen/Interfaces

3.2 Java Enterprise Edition (Java EE) (ca. 5 Wochen)

Java EE Review

Java Database Connectivity (JDBC)

- API für den Zugriff auf (R)DBMS
- Typen von JDBC Treibern
- Grundelemente der JDBC API
- Abfrage- und Änderungsoperationen
- Transaktionssteuerung

Java Persistence API (JPA)

- Object-Relational Mapping (ORM)
- JPA Provider
- ObjectDB
- Grundelemente von JPA
- Konfiguration von JPA
- Annotationen
- Lazy Loading mit FetchType
- Zugriff per Entity-Manager
- Java Persistence Query Language
- Konfiguration in persistence.xml

Unit Tests

- Tests auf der Ebene von Codebausteinen
- Unit-Test Frameworks (z.B. JUnit)
- Testabdeckung
- Simulation durch Mock-Up-Objekte

RESTful Web Services mit JAX-RS

- Was sind Web Services?
- HTTP Request-Typen
- GET/PUT/POST/DELETE
- JavaScript Object Notation (JSON)
- Implementierung von REST Services

Enterprise Java Beans (EJBs)

- Software-Komponenten für Business-Logik
- EJBs als Java-Klassen
- Dienste eines EJB Containers
- Transaktionssteuerung / verteilte Transaktionen
- EJB Container
- Entity Beans, Session Beans und MDBs
- Stateless, Stateful und Singleton EJBs
- JMS, verknüpfbar mit IBM MQ
- Dependency Injection (Inversion of Control)
- Java Transaction API (JTA) und XA Transaktionen
- Verteilte Transaktionen und Transaktionsmonitore
- Container- oder Bean-managed Transaktionen

Java Message Services (JMS)

- MoM Unterstützung für Java
- JMS Provider
- Queues und Publish/Subscribe (Topics)
- Versenden von Messages
- Aufbau einer Message (Header und Body)
- ConnectionFactory Objekt
- Polling von Queues
- MDBs als Listener für Queue / Topic
- MDB Konfiguration (server.xml)

Java Batch Verarbeitung

- JSR 352
- Standardisierte API
- Liberty Implementierung
- Chunk-Verarbeitung und Batchlet
- REST-API oder WOLA
- Job Specification Language (JSL) in XML implementiert
- Batch Interfaces
- ItemReader, ItemProcessor, ItemWriter