



Software Developer Java CDT

Zielgruppe:

Personen, die als Softwareentwickler unter Verwendung der Programmiersprache Applikationen und Applets erstellen wollen.

Voraussetzungen:

Akademiker/-innen der verschiedenen Fachrichtungen, Softwareentwickler/-innen, Informatiker/-innen und Fachinformatiker/-innen. CDT-Eignungstest für IT-Berufe „geeignet“. Ausnahmen sind in Absprache mit dem Kostenträger möglich.

Ziel:

Vermittlung umfangreicher und fundierter Kenntnisse zur objektorientierten Softwareentwicklung, zur Softwareentwicklung mit UML und zur Entwicklung von Applikationen und Applets mit der Programmiersprache Java.

Methode:

Vortrag und praktische Übungen.

Inhalte:

Software Engineering (1): Objektorientierte Programmierung
Objektorientierte Programmierung mit Java
Software Engineering (2): Softwareentwicklung mit UML

Zertifizierungen/ Zeugnis:

SCPJ (Sun Certified Programmer for the Java Platform, SE 6)
Zeugnis von CDT

Hardware:

Server Intel Core 2 Duo, 1,8 GHz, 2 GB RAM, 160 GB HD
Workstations Intel Core 2 Duo, 1,8 GHz, 2 GB RAM, 160 GB HD

Software:

Windows XP Professional
Windows Server 2003 Professional
MS SQL Server 2008
Eclipse IDE for Java Developers

Dauer:

16.08.2010 bis 29.10.2010 (462 Unterrichtsstunden)
(detaillierte Informationen, siehe Rückseite)

BKZ: 7742-103 Maßnahmenr.: 735/2056/2010
AZWW-zertifiziert durch Cert-IT Berlin

Strukturierte
Programmierung

Darstellungs-, Entwurfsmittel; Daten, Strukturen; Programmanweisungen; Algorithmen; Unterprogramme; Funktionen; Module; Dateiverarbeitung, Drucken

Software Engineering (1):
Objektorientierte Software-
entwicklung

Grundlagen Objektorientierung: Objekte, Klassen, Attribute, Methoden, Nachrichten, Assoziation, Klassenhierarchie, (Mehrfach)Vererbung; Polymorphie; dynamisches Binden, Überladen; OOA-, OOD-, OOP-Konzepte; UML (Überblick)

Objektorientiert programmieren
mit Java (1)

Java Development Kit installieren; Elemente eines Java-Programms; Operatoren; Anweisungen; Kontrollstrukturen; Objektorientierung; Strings; Exceptions

Objektorientiert programmieren
mit Java (2)

Java-Programme strukturieren; Pakete verwenden; Collections verwenden; Utility-Klassen einsetzen; Character- und Byte-Streams; Zugriff auf Dateien

Objektorientiert programmieren
mit Java (3)

Multithreading; Event-Handling; Serialisierung; Datenbankzugriff mit JDBC; Grundlagen der Grafik-/Textausgabe; GUI-Dialoge; AWT-Dialogelemente; Bitmaps, Animationen; Grafikprogrammierung mit Swing; Java-Applets

Software Engineering (2):
Softwareentwicklung mit UML 2

Klassen und Beziehungen, verschiedene Arten von Klassendiagrammen; Architektur und Komponenten, verschiedene Arten von Diagrammen; Verhalten