BC401

ABAP Objects

SAP NetWeaver

Course Outline

Course Version: 99
Course Duration: 5 Day(s)
Publication Date: 2014
Publication Time:

Copyright

Trademarks

Disclaimer

THESE MATERIALS ARE PROVIDED BY SAP ON AN "AS IS" BASIS, AND SAP EXPRESSLY DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR APPLIED, INCLUDING WITHOUT LIMITATION WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THESE MATERIALS AND THE SERVICE, INFORMATION, TEXT, GRAPHICS, LINKS, OR ANY OTHER MATERIALS AND PRODUCTS CONTAINED HEREIN. IN NO EVENT SHALL SAP BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR PUNITIVE DAMAGES OF ANY KIND WHATSOEVER, INCLUDING WITHOUT LIMITATION LOST REVENUES OR LOST PROFITS, WHICH MAY RESULT FROM THE USE OF THESE MATERIALS OR INCLUDED SOFTWARE COMPONENTS.

Contents

Course Overview

This course provides a comprehensive and detailed introduction to the programming techniques and the associated syntax of the ABAP Objects language. The only information that is not covered here belongs to separate subject areas like dialog programming or database access, which are discussed in detail in other courses.

This main objective of this course is to provide an introduction to the concepts and programming techniques of object-oriented development with ABAP Objects. This also includes an introduction to modeling.

Furthermore, you will become familiar with the tools in the ABAP Workbench that are used for object-oriented programming with ABAP Objects.

In another part of the course, you will be introduced to programming techniques that sometimes use object-oriented elements of ABAP Objects but can also be used in purely procedural contexts.

SAP's technical component, the **SAP NetWeaver Application Server**, referred to in this course as **SAP NW AS** for short, is embedded in the **SAP NetWeaver** solution and therefore provides a technical basis for almost all SAP solutions. This means that you can use the knowledge and skills you gain from this course not only for developing within **SAP ERP** but also in all solutions that have been developed with ABAP Objects.

Most of the programming techniques, tools, and syntax elements in this course are already available as of *SAP Web AS 6.10*. Only a few concepts were introduced in later releases or enhancement packs. Wherever such concepts are discussed in this course material, we tried to mention the release or enhancement pack as off which they are available.

Target Audience

This course is intended for the following audiences:

- Development project managers
- ABAP Objects developers

Course Prerequisites

Required Knowledge

- Programming experience
- SAPTEC (SAP NetWeaver: Application Platform Fundamentals)
- BC400 (ABAP Workbench: Foundation)

Course Goals

This course will prepare the participant to:

- Understand the details of procedural and object-oriented syntax elements in ABAP Objects and use these effectively
- Create object-oriented models of business applications
- Use the tools in the *ABAP Workbench* to develop object-oriented applications

Course Objectives

After completing this course, the participant will be able to:

- Use fundamental elements of object-oriented modeling in UML
- Create ABAP Objects programs that contain all useful object-oriented programming techniques
- Use the relevant tools to create object-oriented Repository objects
- Describe and exploit the range of applications of ABAP Objects
- Define, raise, and handle class-based exceptions
- Query type and class attributes at runtime

Unit 1

Introduction to Object-Oriented Programming

Unit Overview

This unit will introduce you to the basics of object-oriented software development. The first lesson is an introduction to the new ways of thinking and the related concepts.

The second lesson is a compact introduction to modeling, the step in the software development process that immediately proceeds the actual programming. This will be demonstrated using the modeling standard UML. To begin with, you will only see the most basic and important elements. The lessons will successively build upon this information, introducing the object-oriented programming concepts and the related UML notations in parallel. Therefore, this course will simultaneously teach you object-oriented modeling and programming.

The contents of the first two lessons are essentially also applicable to other modern object-oriented languages. As of the third lesson, syntax elements that are specific to ABAP Objects are introduced. You will need to learn many of these syntax elements in short succession.

Most of the content of the later unit deals with concepts that are entirely new to you. The syntax will play a much smaller part in these units.

Lesson: The Object-Oriented Programming Model

Lesson Objectives

After completing this lesson, the participant will be able to:

- Explain the differences between procedural and object-oriented programming models
- List the advantages of the object-oriented programming model

Lesson: Analysis and Design with UML

Lesson Objectives

After completing this lesson, the participant will be able to:

- Name the most important diagram types in UML
- Create simple class diagrams
- Create simple object diagrams
- Describe sequence diagrams

Lesson: Fundamental Object-Oriented Syntax Elements

Lesson Objectives

After completing this lesson, the participant will be able to:

- Define classes
- Generate and delete objects
- Access attributes
- Call methods

Unit 2

Object-Oriented Concepts and Programming Techniques

Unit Overview

This unit deals with the basic programming techniques common to all object-oriented languages. As far as these concepts are concerned, the only difference between ABAP Objects and other languages like Java or C++ is the syntax.

Every concept exists for a specific reason. It is important that you understand them **all** without exception so that you can use each of them effectively later on. You can only capitalize on the strengths of object-oriented programming if you use all of the concepts in the intended manner.

Lesson: Inheritance and Casting

Lesson Objectives

After completing this lesson, the participant will be able to:

- Define inheritance relationships between classes
- Redefine methods
- Create up-cast assignments (Widening Cast)
- Create down-cast assignments (Narrowing Cast)
- Explain the concept of polymorphism with reference to inheritance
- Use cast assignments with inheritance to make generic calls

Lesson: Interfaces and Casting

Lesson Objectives

After completing this lesson, the participant will be able to:

- Define and implement interfaces
- Implement interface methods
- Use interface references to make up-cast assignments
- Use interface references to make down-cast assignments
- Explain the term polymorphism with reference to interfaces
- Use cast assignments with interfaces to make generic calls

Lesson: Events

Lesson Objectives

After completing this lesson, the participant will be able to:

- Define and trigger events
- Handle events
- Register and deregister event handling
- Explain the key differences between explicit method calls and event-controlled method calls

Unit 3

Object-Oriented Repository Objects

Unit Overview

The first lesson will introduce you to the *Class Builder*, a tool from the *ABAP Workbench* that you probably have not used before. You will focus on learning how to apply the techniques that you learned for local classes and interfaces to global classes and interfaces. A focal point in this lesson is the description of how object-oriented ABAP programming can be used in various ways, as illustrated using the example of the SAP Grid Control and BAdIs.

Lesson: Global Classes and Interfaces

Lesson Objectives

After completing this lesson, the participant will be able to:

- Describe the functions of the Class Builder
- Create global classes using the Class Builder
- Create interfaces using the Class Builder
- Reference global classes and interfaces in other Repository objects
- Create a simple ALV Grid Control
- React to a double-click on an ALV Grid Control with a handler method
- Explain the fundamentals of the BAdI technology and implement a BAdI

Lesson: Editing Global Classes in ABAP Development Tools (ADT)

Lesson Objectives

After completing this lesson, the participant will be able to:

- Know what ABAP Development Tools is
- Create a project in ADT
- Develope a class in ADT
- Use refactoring functions of ADT

Unit 4

Exception Handling and Unit Testing

Unit Overview

This unit is all about robustness of your coding:

In the first lesson you will learn how to define, raise and handle class based exceptions to ensure a well-defined behavior even in error situations.

In the second lesson you will learn how to use ABAP unit tests to verify the correct behavior of your program.

Lesson: Exception Handling in ABAP Objects

Lesson Objectives

After completing this lesson, the participant will be able to:

- Create exception classes
- Raise class-based exceptions in ABAP Objects programs
- Propagate class-based exceptions in ABAP Objects programs
- Handle class-based exceptions in ABAP Objects programs
- Map class-based exceptions in ABAP Objects programs to each other

Lesson: Unit Testing

Lesson Objectives

After completing this lesson, the participant will be able to:

- Describe the concept of ABAP Unit testing
- Name the advantages of ABAP Unit testing
- Understand test classes and test methods
- Perform an ABAP Unit test

Unit 5

Special Object-Oriented Programming Topics

Unit Overview

Advanced object models often use so-called "patterns". In the first lesson of this unit you will get to know the most important of those patterns - along with some further concept and language features of ABAP Objects which are necessary to implement these patterns.

In the second lesson you will learn how to use the Run-Time Type Services (RTTS) to analyze the properties of objects and classes at run time.

The final lesson demonstrates the concept of persistent objects and illustrates how a completely object oriented ABAP program could look like.

Lesson: More Object-Oriented Techniques and Patterns

Lesson Objectives

After completing this lesson, the participant will be able to:

- Define abstract classes
- Define abstract methods
- Define final classes
- Define final methods
- Understand the special use of public read-only attributes
- Understand the term "navigation method"
- Chain functional methods
- Limit the visibility of the constructor
- Understand the term "factory method"
- Explain the "singleton pattern"
- Define friendship relationships between classes

Lesson: RTTS (Run Time Type Services)

Lesson Objectives

After completing this lesson, the participant will be able to:

- Query type attributes at runtime

- Create types dynamically

Lesson: Persistent Objects and OO-Transactions

Lesson Objectives

After completing this lesson, the participant will be able to:

- Create persistent objects using the standard technique
- Load persistent objects using the standard technique
- Access components of persistent objects
- Create an object-oriented transaction (transaction code type)

Unit 6

Case Study

Unit Overview

Case Study

Lesson: Case Study

Lesson Objectives

After completing this lesson, the participant will be able to:

- Explain the concepts relating to object-oriented programming with ABAP Objects demonstrated in this course and apply these in a small, simple application