

KillTest

Mejor calidad. Mejor servicio



Preguntas de práctica

<https://www.killtest.es>

Actualizaciones gratuitas durante un año

Exam : C_ABAPD_2507

**Title : SAP Certified Associate -
Back-End Developer -
ABAP Cloud**

Version : DEMO

1. When defining a METHOD, which parameter type can only have 1 value?

- A. IMPORTING
- B. EXPORTING
- C. CHANGING
- D. RETURNING

Answer: D

Explanation:

In ABAP Object-Oriented Programming within ABAP Cloud, methods can define multiple parameters of type IMPORTING, EXPORTING, or CHANGING. However, for RETURNING parameters, only one value is permitted per method.

This restriction ensures that RAP BOs and ABAP Cloud classes expose methods with clear, unambiguous outputs, aligning with best practices of encapsulation and functional programming design.

IMPORTING → multiple allowed

EXPORTING → multiple allowed

CHANGING → multiple allowed

RETURNING → exactly one allowed

Verified Study Guide

Reference: ABAP Objects Programming Guide (Methods), ABAP Cloud Back-End Developer Documentation – Method Signature Rules.

2. In a class you use the statement DATA var TYPE ...

What may stand in place of the type? (Select 2 correct answers)

- A. The name of a type defined privately in class ZCL_CLASS_A
- B. The name of a domain from the ABAP Dictionary
- C. The name of a type defined privately in another class
- D. The name of a data element from the ABAP Dictionary

Answer: A, D

Explanation:

The ABAP DATA statement declares a variable with an assigned type.

A. Private type in the same class (ZCL_CLASS_A) → Allowed. A class can use its own local type definitions, declared using TYPES.

B. Domain from ABAP Dictionary → Not allowed directly. Domains define technical attributes but cannot be referenced directly in DATA; they must be wrapped in a data element.

C. Type defined privately in another class → Not accessible, since private definitions are encapsulated.

D. Data element from ABAP Dictionary → Allowed, because data elements are global dictionary objects.

This follows ABAP Cloud extensibility rules, ensuring encapsulation and strict typing.

Verified Study Guide

Reference: ABAP Dictionary Development Guide, ABAP Cloud Back-End Developer Learning Material (Variable Typing and Encapsulation).

3. After you created a database table in the RESTful Application Programming model, what do you create next?

- A. A data view
- B. A service definition

- C. A metadata extension
- D. A projection view

Answer: D

Explanation:

In the ABAP RESTful Application Programming Model (RAP), development follows a bottom-up approach. The sequence starts with the database table (persistence), followed by creating a data model view (interface view), and then a projection view.

The projection view is the next artifact after the database table because it exposes only the fields needed for the app, aligning with the encapsulation principle of RAP. This ensures that only the required subset of the underlying data model is made available.

Verified Study Guide

Reference: ABAP RAP Development Guide – Data Modeling and Projection Views.

4.What are some principles of encapsulation? (Select 2 correct answers)

- A. Attributes can be changed through public class methods.
- B. Attributes can be changed by the client program directly.
- C. Attributes cannot be changed.
- D. Attributes can only be changed by the class.

Answer: A, D

Explanation:

Encapsulation in ABAP OO and ABAP Cloud ensures that internal object details are hidden from outside consumers.

A. Attributes can be changed through public methods → Correct, because controlled access is provided through getter/setter or other methods.

B. Attributes can be changed by the client program directly → Incorrect, this violates encapsulation.

C. Attributes cannot be changed → Incorrect, they can be changed, but only via allowed mechanisms.

D. Attributes can only be changed by the class itself → Correct, ensuring business logic consistency.

This aligns with RAP behavior definitions (BDEF) where internal attributes are encapsulated and only manipulated through behavior implementation methods.

Verified Study Guide

Reference: ABAP Objects Programming Guide – Encapsulation Principles.

5.Given the following ABAP code, which exception will be raised on execution?

```
CONSTANTS c_char TYPE c LENGTH 1 VALUE ' '.
```

```
TRY.
```

```
result = 2 / c_char.
```

```
out->write( |Result: { result }| ).
```

```
CATCH cx_sy_zerodivide.
```

```
out->write( |Error: Division by zero is not defined| ).
```

```
CATCH cx_sy_conversion_no_number.
```

```
out->write( |Error: { c_char } is not a number!| ).
```

```
CATCH cx_sy_itab_line_not_found.
```

```
out->write( |Error: Itab contains less than { 2 / c_char } rows| ).
```

```
ENDTRY.
```

- A. cx_sy_zerodivide
- B. cx_sy_conversion_no_number
- C. cx_sy_itab_line_not_found

Answer: B

Explanation:

Here, c_char is defined as a character type with a space ' ' as its value.

When attempting 2 / c_char, ABAP tries to interpret the character ' ' as a number. Since it is not a numeric value, ABAP raises the conversion error cx_sy_conversion_no_number.

cx_sy_zerodivide would occur only if the denominator was zero numeric.

cx_sy_itab_line_not_found applies to internal table access errors, not relevant here.

This is consistent with ABAP Cloud runtime exception handling, where strict typing and error categories are clearly defined.

Verified Study Guide

Reference: ABAP Keyword Documentation – Exception Classes in Arithmetic Operations.