Object Oriented Programming

SAP ABAP Class

What is Class?

A class is a collection of equivalent properties and set of instructions. Class is a template to objects and define objects. In technical, objects are runtime instances of a class.

Any number of objects can be created on a single class. Each instance/object of a class has a unique identity and its own set of values for its attributes.

The data functions within the class are called as class members. The object attributes are defined by the class components that describes the state and behavior of objects.

The below statements define the structure of a class -

- A class contains components.
- Each component is assigned to a visibility section.
- Classes implement methods.

Class Types -

ABAP Classes are two types based on their declaration and visibility. Those are -

Class	Description
Global Classes	Global classes are accessed by all ABAP programs. Global classes are defined in the Class Builder (Transaction SE24) in the ABAP Workbench. Global classes are stored centrally in **Repository class library**

Class	Description
	class pools.
Local Classes	Local classes are defined within ABAP program. Local classes can only use in the program where they are defined and not available to other program. Local classes consist of ABAP source code and should code in between the **CLASS** and **ENDCLASS**.

The system first searches for a local class with the specified name when a class in an ABAP program. If it does not find any local classes, then looks for a global class. There is no difference between using a global class or a local class apart from visibility.

Class Definition -

The full class definition consists a declaration part and an implementation part if required. The class declaration syntax is -

```
CLASS <class-name> DEFINITION.
        …
        Statements-block
        …
ENDCLASS.
```

It contains the declaration for all class components (attributes, methods, events). The local class declaration should place at the beginning of the program as it belongs to the global program data.

To declare methods in the class declaration, implementation part should be coded. The syntax for method declaration is -

```
CLASS <class-name> IMPLEMENTATION.
        …
        Statements-block
        …
ENDCLASS.
```

The class implementation part contains the all methods implementation of the class.

Class Components -

The class components create its contents. All components are declared in the class declaration part and the components define the object attributes in a class. All the class components are visible within the class.

There are two types of class components – instance components (those exist separately for each object in the class which are instance specific) and static components (those exist only once for the whole class regardless of the number of instances).

Classes can define the below components in ABAP objects -

Class Components	Description
Attributes	Attributes are internal data fields within a class that are declared with any ABAP data type such as C, I, F and N. The object state is determined by its attributes contents.
Methods	Methods are internal procedures in a class and defines the behavior of an object. Methods can access all class attributes that allows them to change the object data content.
Constructors	Constructors are special methods that are called automatically while creating an object or instantiated (accessing the class components) first time. Constructor gets triggered when an object is created.
Events	Events are functions that are triggered based on the result of a condition. Events are used by Objects or classes to trigger event handler methods in other objects or classes.
Types	User-defined ABAP data types defined with a class using the TYPES statement. Types are not instance-specific. Type exist only once for all the class objects.

Class Components	Description
Constants	Constants are special static attributes. Constants can declare by using the CONSTANTS statement. Constants are not instance-specific. Constants exist only once for all the class objects.

Class Attributes:

Attributes are internal data fields within a class that can be declared with any ABAP data type such as C, I, F and N. The object state is determined by its attributes contents.

Attributes are two types and those are -

Attribute Types	Description
Instance Attributes	The instance attributes contents defines the instance-specific state of an object. The states are different for different objects. Instance attributes can be declared using the DATA statement.
Static Attributes	The static attribute defines the state of the class. The states are common for all different object instances of the class. Static attributes exist only once. Static attributes can declare by using the CLASS-DATA statement. All the objects in a class can access its static attributes. If a static attribute in an object changed, the change is visible in all other objects of the class.

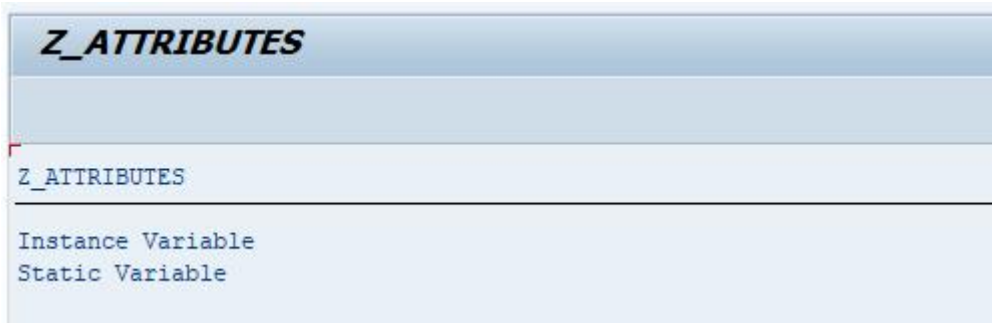
Example -

Simple example to define instance variable and static variables.

Code -

```
REPORT  Z_ATTRIBUTES.
* Class and Method definition
CLASS classnew DEFINITION.
PUBLIC SECTION.
   * Defining instance variable inst_var
  DATA inst_var(30) TYPE C VALUE 'Instance Variable'.
  * Definging static variable stat_var
  CLASS-DATA stat_var(30) TYPE C VALUE 'Static Variable'.
  * Defining public method methodnew for the class classnew
  METHODS: methodnew.
ENDCLASS.
* Class & Method Implementation
CLASS classnew IMPLEMENTATION.
* Method implementation
  METHOD methodnew.
         * Displays instance variable
    WRITE:/ inst_var.
    * Displays static variable
    WRITE:/ stat_var.
  ENDMETHOD.
ENDCLASS.
START-OF-SELECTION.
* Defining object of reference type to class.
DATA: objectnew TYPE REF TO classnew.
* Creatng class object
CREATE OBJECT objectnew.
* Call method using object.
CALL METHOD: objectnew->methodnew.
```

Output -



Explaining Example -

In the above example, each and every statement is preceeded with a comment to explain about the statement. Go through them to get clear understanding of example code.

inst_var is instance variable because it was defined by using DATA clause.
Stat_var is static variable because it was defined by using CLASS-DATA clause.

Methods:

Methods are internal procedures in a class that defines the behavior of an object. Methods can access all class attributes that allows to change the object data content.

Methods have a parameter interface that is used to supply values while calling methods and receive values back from methods. The method definition can code with class definition and method implementation can code in class implementation.

The method implementation syntax is -

```
METHOD <method-name>.
        ...
        Statements-block
        ...ENDMETHOD.
```

Methods can declare by using the CALL METHOD statement. Method statements-block should code in between METHOD ENDMETHOD.

Methods are three types and those are -

Class Methods	Description
Instance Methods	Instance methods are very specific to instance. Instance methods can declare using the METHODS statement. Instance methods can access all the class attributes of a class and can trigger all the class events.
Static Methods	Static methods can be declared using the CLASS-METHODS statement. Static methods can only access static attributes and trigger static events.
Special Methods	There are some special methods based on their definition and usage. When we first call a method using CALL METHOD, there are two special methods **constructor** and **class_constructor** that are automatically called.

Example -

Simple example to define, call instance and static methods.

Code -

```
REPORT  Z_METHODS.
* Class & Methods Definition
CLASS classnew DEFINITION.
PUBLIC SECTION.
* Instance method definitions
  METHODS instmethod.
*static method definition
  CLASS-METHODS statmethod.
ENDCLASS.
```

```
* Class implementation
CLASS classnew IMPLEMENTATION.
* Instance method implementation
  METHOD instmethod.
    WRITE:/ 'Executing instance method'.
  ENDMETHOD.
* Static method implementation
  METHOD statmethod.
    WRITE:/ 'Executing static method'.
  ENDMETHOD.
ENDCLASS.

START-OF-SELECTION.
* Defining object objectnew for the class classnew
DATA: objectnew TYPE REF TO classnew.
* Creating object objectnew
CREATE OBJECT objectnew.
* Calling instance method, static method
CALL METHOD: objectnew->instmethod,
             objectnew->statmethod.
```

Output -

Explaining Example -

In the above example, each and every statement is preceeded with a comment to explain about the statement. Go through them to get clear understanding of example code.

instmethod is instance method as it was defined by using METHODS clause.
statmethod is static method as it was created by using CLASS-METHODS clause.

Constructors

Constructors are special methods that are called automatically while creating an object or instantiated (accessing the class components) first time. Constructor gets triggered when an object is created.

To trigger a method, we need to call a method. CONSTRUCTORS are two types and those are -

Constuctor Types	Description
Instance constructor	Instance constructor methods can only have importing parameters and there will not be any exporting parameters. The name of the CONSTRUCTOR method must be CONSTRUCTOR only.
Static constructor	Static constructors are mainly used to set the default values globally irrespective of instances/methods. These methods will not have any importing and exporting parameters. Static constructor methods gets executed only once. The name of the CLASS CONSTRUCTOR must be CLASS_CONSTRUCTOR.

Example -

Simple example to explain how the constructor is being triggered.

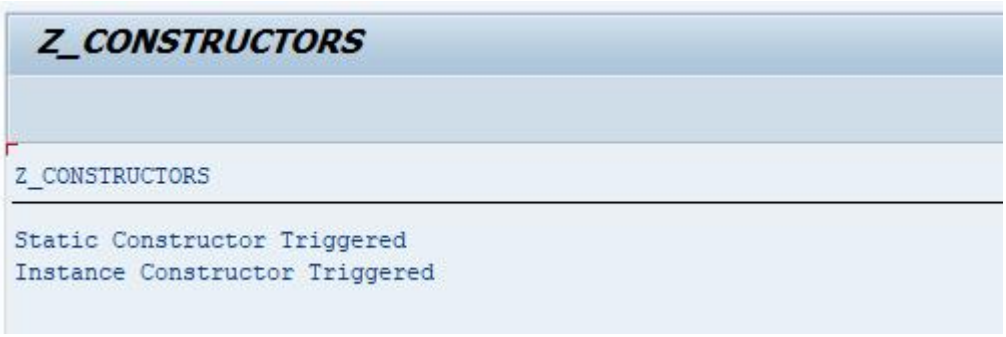
Code -

```
REPORT  Z_CONSTRUCTORS.
* Class & methods definition
```

```
CLASS classnew DEFINITION.
PUBLIC SECTION.
* Instance constructor definition
  METHODS constructor.
* Static constructor definition
  CLASS-METHODS class_constructor.
ENDCLASS.
* Class & method implementation
CLASS classnew IMPLEMENTATION.
* Instance method implementation
  METHOD constructor.
     WRITE:/ 'Instance Constructor Triggered'.
  ENDMETHOD.
* Static method implementation
  METHOD class_constructor.
     WRITE:/ 'Static Constructor Triggered'.
  ENDMETHOD.
ENDCLASS.

START-OF-SELECTION.
* Defining object objectnew for the class classnew
DATA: objectnew TYPE REF TO classnew.
* Creating object objectnew
CREATE OBJECT objectnew.
```

Output -



Explaining Example -

In the above example, each and every statement is preceeded with a comment to explain about the statement. Go through them to get clear understanding of example code.

constructor is instance constructor as it was defined by using METHODS clause. **class_constructor** is static constructor as it was created by using CLASS-METHODS clause.

The static and instance constructors are called when the object objectnew is created. Static constructor is the first constructor triggered before and instance constructor triggered.

ABAP Objects

Objects establish the key concept in object oriented programming. An object is a self-contained unit that has its own status, behavior and identity. Objects are patterns or instances of classes. An object has the following three main characteristics −

- State
- Behaviour
- Identity

The object **behavior** is determined by its methods and refers to the changes that occur in its attributes over a period. Each object has a unique identity that can be used to distinguish it from other objects. Two objects may exhibit the same behaviour however those may or may not have the same state and they never have the same identity. The object behaviours or methods represents the object performed actions. Objects creates a container that contain data which combines the character to the respective behavior to provide services. An object is accessed referring via reference variables to the address.

The object **identity** is determined by the address in memory. Likewise, the object identity never change throughout its lifetime. Objects can interact with another object by sending messages. An object in a program that performs a

certain tasks that should reflect a real object of the task. A clear difference can be made between the public, private and protected components with objects.

The object **state** is described as a set of attributes and their values. For example, a bank account has a set of attributes such as Account Number, Name, Account Type, Balance.

One object can interact with another by accessing its attributes directly in a method, or calling methods, or triggering an event.

The services are known as methods also known as operations or functions. The attributes of an object can't be changed directly by the user, however those can changed only by the methods of the object.

In programming, Objects can be used as user-defined data type with help of class and contain code to manipulate the data. Objects are called as variables of the type class and we can create any number of objects belongs to specific class after defining it.

Creating an Object -

Creating an object is of two types based on how it is defined. Those are -

- Creating a reference variable to the class. The syntax is −

DATA: <obj_name> TYPE REF TO <class_name>.

- Creating an object from the reference variable. The syntax is −

CREATE Object: <obj _name>.

-

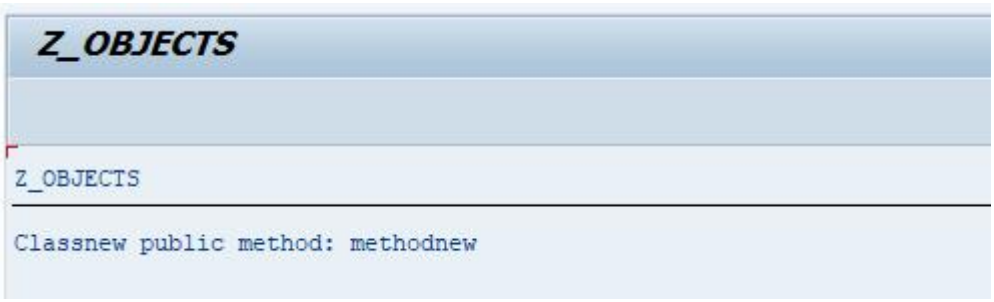
Example -

Simple example to create class object.

Code -

```
REPORT  Z_OBJECTS.
* Class definition
CLASS classnew DEFINITION.
PUBLIC SECTION.
* Defining Method
  METHODS: methodnew.
ENDCLASS.
*Class implementation
CLASS classnew IMPLEMENTATION.
* Method Implementation
  METHOD methodnew.
     Write:/ 'Classnew public method: methodnew'.
  ENDMETHOD.
ENDCLASS.
START-OF-SELECTION.
* Defining a class object
DATA: objectnew TYPE REF TO classnew.
* Creating object objectnew
CREATE OBJECT objectnew.
* Calling method using object.
CALL METHOD: objectnew->methodnew.
```

Output -



Explaining Example -

In the above example, each and every statement is preceeded with a comment to explain about the statement. Go through them to get clear understanding of example code.

classnew is the class defined with a public method **methodnew**. The **methodnew** implementation coded in **classnew** implementation. **objectnew** declared as of type reference to class **classnew**.

CREATE OBJECT creates the **objectnew** object. **objectnew->methodnew** calls the method **methodnew** and executes the code in method implementation.