ABAP COMPLETE OOP + INTERNAL TABLES MEGA CHEAT SHEET

===

SECTION 1 — OOP BASICS

----

1. CLASS & OBJECT

```
CLASS zcl_person DEFINITION.
PUBLIC SECTION.
DATA name TYPE string.
METHODS constructor IMPORTING i_name TYPE string.
METHODS show RETURNING VALUE(out) TYPE string.
ENDCLASS.

CLASS zcl_person IMPLEMENTATION.
METHOD constructor.
name = i_name.
ENDMETHOD.

METHOD show.
out = |Name: { name }|.
ENDMETHOD.
ENDCLASS.
```

Usage:

```
DATA(lo) = NEW zcl_person( i_name = 'Ana' ).
out->write( lo->show( ) ).
```

2. OBJECT REFERENCE vs DATA REFERENCE

Object Reference → points to an OBJECT (instance of a class)

Data Reference → points to a VARIABLE or TABLE or STRUCTURE

```
DATA lo_person TYPE REF TO zcl_person. "object reference
CREATE OBJECT lo_person.

DATA lr_data TYPE REF TO ty_structure. "data reference
GET REFERENCE OF ls_struct INTO lr_data.
```

3. INHERITANCE

```
CLASS parent DEFINITION.
PUBLIC SECTION.
METHODS speak.
ENDCLASS.

CLASS child DEFINITION INHERITING FROM parent.
PUBLIC SECTION.
METHODS speak REDEFINITION.
ENDCLASS.
```

4. POLYMORPHISM

```
DATA lo_parent TYPE REF TO parent.
lo_parent = NEW child( ).
lo_parent->speak( ). "child version runs
```

5. INTERFACES

```
INTERFACE if_person.
METHODS display.
ENDINTERFACE.

CLASS zcl_worker DEFINITION.
PUBLIC SECTION.
INTERFACES if_person.
ENDCLASS.
```

6. CASTING (UP & DOWN)

Upcast:

```
DATA lo_parent TYPE REF TO parent.
lo_parent = NEW child( ).
```

Downcast (safe):

```
DATA lo_child TYPE REF TO child.
TRY.
lo_child ?= lo_parent.
CATCH cx_sy_move_cast_error.
```

ENDTRY.

7. FACTORY METHOD

CLASS zcl_factory DEFINITION.

PUBLIC SECTION.

CLASS-METHODS create_person RETURNING VALUE(obj) TYPE REF TO zcl_person.

ENDCLASS.

CLASS zcl_factory IMPLEMENTATION.

METHOD create_person.

obj = NEW zcl_person(i_name = 'Default').

ENDMETHOD.

ENDCLASS.

Usage:

DATA(lo) = zcl_factory=>create_person( ).

SECTION 2 — INTERNAL TABLES

----

1. TABLE TYPES

STANDARD TABLE → unsorted, linear search

SORTED TABLE → sorted key, binary search

HASHED TABLE → hashed key, O(1) access

2. DEFAULT KEY

Default key includes ALL non-numeric fields.

3. FULL TABLE CREATION EXAMPLES

A. Standard Table

DATA it TYPE STANDARD TABLE OF ty.

B. Sorted Table with unique key

DATA it TYPE SORTED TABLE OF ty WITH UNIQUE KEY id.

C. Hashed Table

DATA it TYPE HASHED TABLE OF ty WITH UNIQUE KEY id.

D. Adding Secondary Keys

DATA it TYPE STANDARD TABLE OF ty

WITH NON-UNIQUE SORTED KEY age COMPONENTS age

WITH UNIQUE HASHED KEY id COMPONENTS id.

4. VALUE Constructor Table Fill

----

itab = VALUE #(

(id = 1 name = 'A')

(id = 2 name = 'B')

).

5. FOR LOOP Constructor

----

itab2 = VALUE #(FOR row IN itab (row-id)).

With filter:

itab3 = VALUE #(FOR row IN itab WHERE (id > 50) (row)).

6. LET inside FOR

----

itab = VALUE #(

FOR row IN itab

LET len = strlen(row-name)

IN IF len > 3 THEN row

).

7. TABLE EXPRESSIONS

----

ls = itab[id = 10].

ls = itab[id = 99] OPTIONAL.

8. FIELD SYMBOL READ

READ TABLE itab ASSIGNING FIELD-SYMBOL() WITH KEY id = 10.

IF IS ASSIGNED.

9. DELETE Operations

DELETE itab WHERE city = 'Berlin'.

DELETE ADJACENT DUPLICATES FROM itab COMPARING id.

10. SORT

SORT itab BY id.

11. RANGE TABLE

DATA it_range TYPE RANGE OF i.

it_range = VALUE #(

(sign = 'I' option = 'GT' low = 100)

(sign = 'I' option = 'LT' low = 500)

).

12. CORRESPONDING Mapping

----

dto = CORRESPONDING #(db

MAPPING (first = fname last = lname)

EXCEPT salary).

13. DATA REFERENCES + Tables

DATA lr_tab TYPE REF TO data.

CREATE DATA lr_tab TYPE TABLE OF i.

ASSIGN lr_tab->* TO FIELD-SYMBOL().

APPEND 10 TO .

14. LOOP AT GROUP BY

LOOP AT itab INTO DATA(row)

GROUP BY (city = row-city) INTO DATA(group).

LOOP AT GROUP group ASSIGNING FIELD-SYMBOL().

ENDLOOP.

ENDLOOP.

15. REDUCE

----

total = REDUCE i(INIT x = 0 FOR row IN itab NEXT x = x + row-amount).

END OF CHEAT SHEET