





In this lesson, I want to go through using internal tables with about objects.

Now, the cost up to now has concentrated on, you know, just individual fields and maybe the old structure

here and there, but we haven't used internal table.

Well, the reason for that is there's actually no difference in using internal tables with APAP objects

as there is to normal procedural programming.

Apart from the fact that I mentioned earlier in the course that you can't use the old style of internal

tables with headlines.

OK, so when you use internal tables, create your internal tables like you should have been doing over

the last five, six, seven years, however long you've been programming, but use work areas instead

of headlines now to show you exactly how we can use internal tables and about objects.

I've got an example here and it's really simple.

You should be able to look at it and go, you know what is exactly what I thought he should be when

by taking into account my knowledge of procedural programming.

So let's go through it line by line and then we'll move on.

So first of all, but defining a class in this example, we have the public section.

And to start with, I'm declaring a type and we have a time when we're making a standard table of a

table.

And in this case, I'm referring to an update chinnery table that already exists.

Then I use a data statement to create the internal table based on the type.

Simple as that.

Doesn't get any easier, does it?

In this example, I am then creating a constructor and then when we close off the definition section

of our class and come to the implementation, you can see I've got to construct a method and all I'm

doing is selecting all the records from the database table and popping them into the internal table.

Then we have a method.

Then as an example, in our calling program, we create a reference variable to the class and then we

create the object that is it super simple.

There should be no surprises here.

Decoding should be extremely familiar to you from your procedural programming experience.

Now, to just extend on that a little bit, what I want to do at the end of this video, I'm going to

ask you to do a bit more homework and generate another program.

And I want you to create a program using internal tables and accessing data from a database table.

And for this exercise, I want to use the TSP fly table that everybody should be able to find in their

system, whether it's a normal SAPI system that's in your normal development environment or even if

you're using the developer versions that you can download free of charge from CNN dot com, you should

all be able to find this table in the system.

And if needed, I've got some instructions here to show you how to populate the table with data.

So let me do that, let me bring up the.

SAPI guey, here we are, and what I'm going to do if I just made this a lot smaller, we can see here

I've got some instructions that this table is a standard table and I'm going to go to see 11.

SPF Thalys.

When we view the structure of it.

You can see it's a flight schedule, so within SFP, they've developed this demo system of a flight

booking system so we can book flights to go from one country to the next and so on.

And this table holds the flight schedule.

And if we have a brief look at the data.

I have massive masses of data in this I think this 26 records, you can see we got the clients as usual,

but we have the carrier ID connection I.D. the country from city from, airport from.

And then we have the country to city to airport to flight times, departure times, arrival times,

distance, distance I.D., which is miles and kilometre.

And then we have the flight type and the period field, so it's not a massive table, but this is ideal

for us to work on in this example.

Now, if you don't have any data in your table, in your system, Sappi, have provided an example.

Well, not an example.

They provided a program that will generate data for it.

And this is called Sappi B.C. and discovered data on Disko generator.

And you'll normally find a transaction as well called B.S. underscore data, underscore again, so all

you need to do is run that.

B.C. underscored data underscore Jen.

And then you can see here we have this input screen that gives you some options for you to decide how

many records you want to generate in these example tables.

So if you do have some data in, you can you can delete them to begin with, delete table entries,

but then choose the records that you want.

I chose the standard data record set of 26.

That's all.

I clicked and then click to execute and it generated the records for me where only bothered about this

spelling table were not too bothered about in all these individual flights and bookings data.

Because if you're running, let's say with the monster data record set, it can take ages to run because

it generates a lot of data.

And you can see here actually these options are great out until you run this in the background mode.

So if you do want to generate lots running background mode and you can select these options, if not

just do as I do, do the standard data, set it, execute, it may take a little bit of time to run,

but then you have some data in your table.

So back to the slide.

Scoots across, once you have data in your table, then this is the assignment I've got for you.

It's quite a lengthy one, so it should take you a bit of time to do and the solution that you produce.

We will compare against the solution I produce in a video.

But because there's quite a bit to do here, it's likely your solution is going to differ from mine.

But as long as you get the main concepts, that's what we're focused on.

So when we go through my solution, I'll go through how I create each individual method and so on,

and you can compare it to what you've done.

So first of all, let's go through this assignment step by step so you understand exactly what I'm asking

you to do.

So some basic guidelines.

First of all, the class will have one attribute only.

That's all we require.

And because we're working with internal tables, just in this specific lesson, I want you to make that

one attribute, an internal table.

And this internal table needs to replicate all the data found in ESP fly, so there's 26 records.

All access to the data must be through object methods.

OK, so that means our attribute that we declare has to be in the private section.

Next, I want you to create methods to access all the data in the internal table.

And then create a constructor to fill the internal table from the database.

So these are general guidelines that, you know, you've got to implement that will allow us then to

build these additional methods.

So when we come down to these methods here, we have two, four, six, seven methods in total.

And these are going to take you through creating the different types of methods we have covered in this

course so far.

So first of all, I want you to create a constructor that will load the record from ESP with ESP to

the internal table, just as I mentioned above.

Then we have the old data and all I want for this is for you to write out to the screen all of the data

in our internal table and we have show connection already.

And I want this one to take in a connection ideas, a parameter right out the following information

so you can read that, take notes of it and code it appropriately.

There's a bit of logic.

He is saying if no connection exists right out, an error message followed by a connection ID that is

passed in.

Next, we have no flights to.

This is the number of flights going to a specific airport, so taking an airport code, it could be

you you'll see from the data set, we have codes like FAA for Frankfurt, MSFC for San Francisco, et

cetera, and return the number of flights that travel to that airport.

And another method get connection ID, this method will take in two airport codes, the first one will

represent the departure airport and the second one will represent the arrival airport.

Then I want you to return the connection video of the flight that matches that record.

It's a combination of parameters that we pass to this method actually returns no records at all.

Then I want you to return zero.

Get flight time, this method accepts the connection idea again, and I want you to return the number

of minutes that the flight actually takes.

And the final method gets all connection facts.

This one is to take in a connection ID and return a structure that contains all the information about

that connection.

So as a little note here, this structure should correspond to a row in the table that we define at

the top.

OK, so that is the class assignment.

It's quite high level.

So it's open for you to interpret it as you wish, but it's pretty specific in what I want you to code

up what you should be passing into each method and what you should be returning and writing out to the

screen and so on.

So take your time, study it.

I do think this will likely take you at least an hour.

I would think if you're new to wound up and you've come to this course from the beginning, this guy

may take you a lot longer.

Just trial and error.

Take your time.

Don't try and rush things and just try and code it as best as you can.

Then when you come back, once you've done it, the next video, I will show you how I code it step

by step.