CS315 HOMEWORK 3

Alp Tuğrul Ağçalı

21801799

Section 3

Nested subprograms

A nested function in the Dart programming language is a function that is defined within another function. These nested functions have access to the variables and parameters of the outer function and can be called from within it in the same way as regular functions.

Scope of Local Variables

In the Dart programming language, a local variable is a variable that is only accessible within a certain area of code, known as its scope. These variables are defined within a function or block of code and are not visible or usable outside of that function or block. Local variables are useful for storing temporary values or information that is only needed within a specific area of code.

Parameter Passing

In Dart, you can pass values to functions or methods by using parameters. These values, called arguments, are stored in the parameters, which are variables defined in the function or method. The order of the arguments must match the order of the parameters in the function's definition when the function is called.

There are two main ways to pass parameter to function and those are named and positional parameter passing. Positional parameter passing is looks for its position of the parameters where base function found. Moreover, named parameter passing is specifying parameters with their name. When we call them, we need to write their name with ":" and value of the parameter.

Named parameters will explained below.

Keyword and Default Parameter

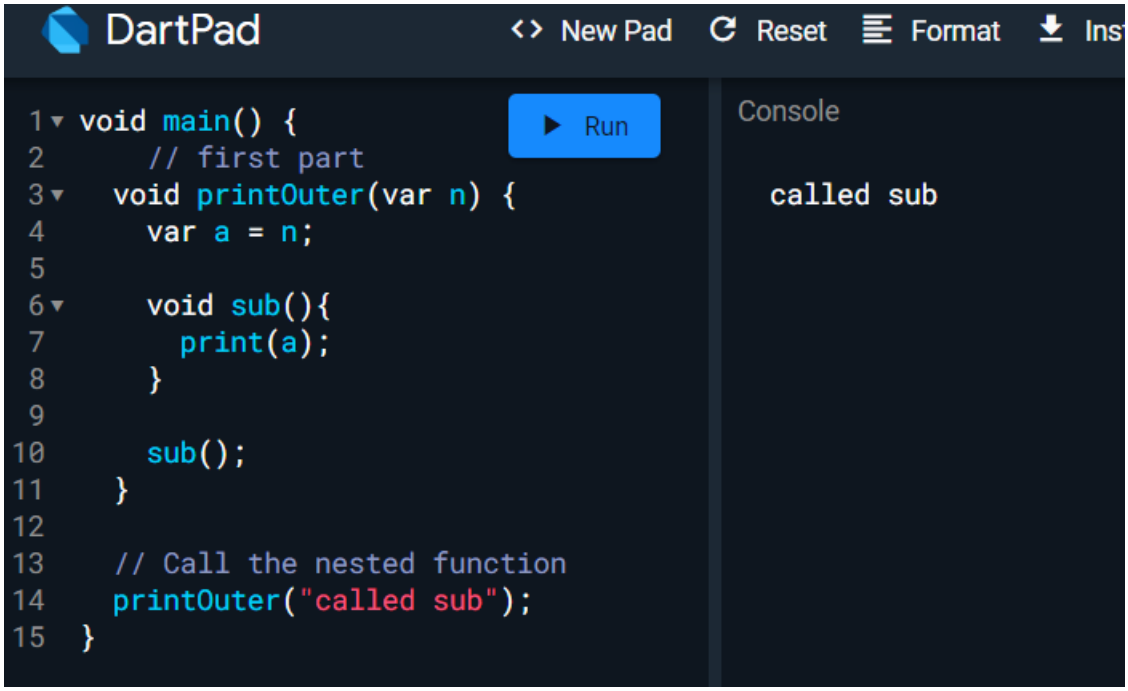
A keyword parameter in the Dart programming language is a function parameter that has a default value and is identified by its name instead of its position in the argument list. This allows you to specify the values for certain parameters when calling a function, while others will use their default values. This procedure called named parameter passing.

Closure

In the Dart programming language, a closure is a function that can be passed around as a value, stored in a variable, or passed as an argument to another function. Closures are often used to create anonymous functions that can be used as a one-time function or as a callback function.

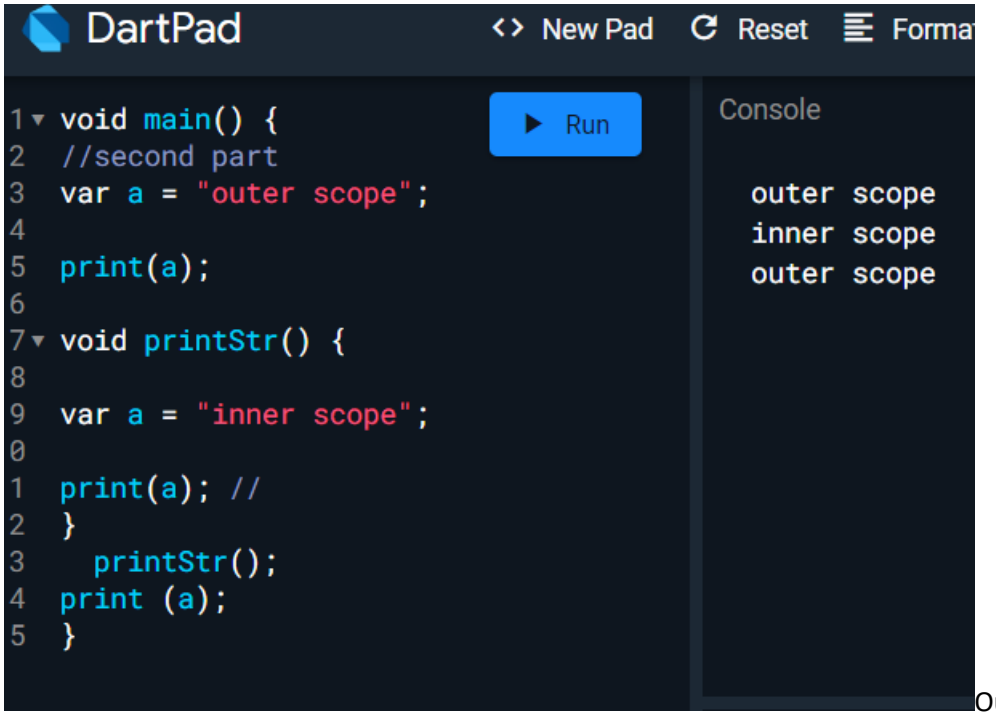
CODES

Nested Sub Program



In this example it is shown that sub programs can be called from outer function and we can use outer function's local variable from inner sub function.

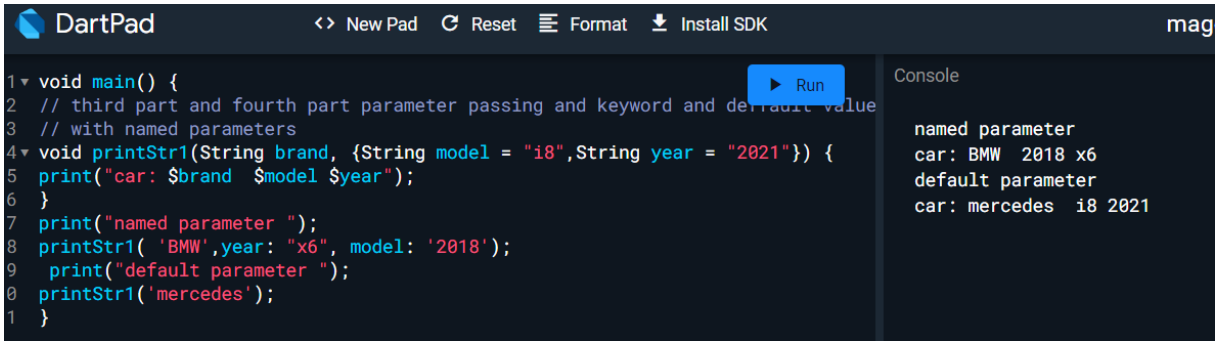
Scope of Local Variables



Output

In this example even we change variable “a” in the inner function then, in the outer function, it remains same. Therefore, it can be seen that change of the inner scope do not affect the outer variable. So, scope of local variable protect it self in that scope even we change it from inner scope.

Parameter Passing, Keyword and Default Value Explained

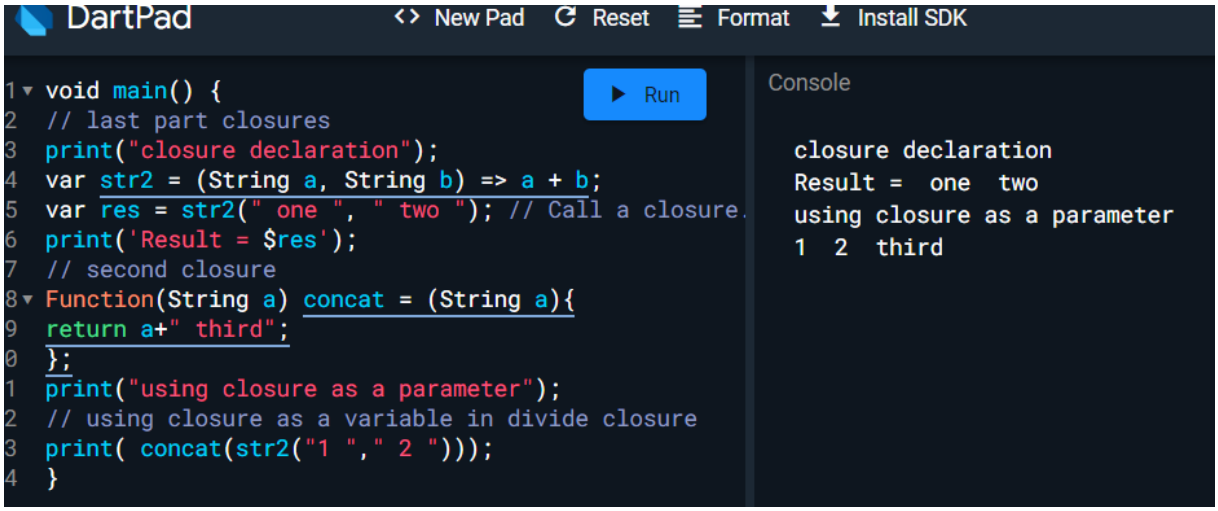


In the named parameters example (First Call of printStr1()) first parameter "brand" has positional parameter passing and second parameter and third parameter have named (default) parameter passing method. Because we call function with their keyword and position of the parameters are not important. We can use those parameters without their knowledge.

In the second example (second call of printStr1()) only first parameter is specified other parameters are not specified and output is with default values.

Therefore, In dart programing if we want to use default value parameter passing method also we need to use keywords and we can use them in both default and named parameter passing methods.

Closures



In this example I create a closure for demonstrating that we can declare functions as variables and these variables acts as a function. Therefore, thanks to closures we can keep functions in the variables.

Also, in the second 4th print statement it can be seen that we can use function as a parameter if we declare them as a closure.

Therefore, one of the main advantages of closure is using functions as a parameter in other functions or closures.

Evaluation Of Language

Writability: Dart programing language has good writability because we can use all of features like other common programming languages and understanding of code when writing is simple. Especially for this homework thanks to closures writability is enhanced because we can use functions as a parameter, so it decreases the line of code. If we try to write same codes in other languages which has not have a closure features, it causes more lines of code. And for parameter passing this language allows us to write both positional and default parameters also when we declare them as a default, we need to specify a name for those. Therefore, it can be used as a named parameter. So, it is easy to write because only changes are using name we do not need to know its position of the named parameter.

Readability: Closures decreases the readability because following code is difficult in closures because for closures if use them as a parameter, we need to know its return type and value of return maybe it decreases lines of code but understanding of code not simple.

From my point of view Dart language is the best language because in print statements we do not need to specify variables position only need is putting dollar assign. Moreover, declaring a variable is so easy we do not need to specify its type. Therefore, it's the best language for me.

Learning Strategy

For this homework, I use text book for clarify some topics such as closure and I used stackoverflow web site for syntax of dart language.

https://dartpad.dev/

https://stackoverflow.com/