Assignment – 1

Name: Muhammad Umair Khalid.

Batch: Android (Kotlin)

Submitted To: Kashif Mehmood.

Date: 4th July 2024.

Problem Statement:

Initially you are given a URL having placeholders in it; represented by **{}.** We are also given a set of variables that hold the values that will go inside these place holders. The numbers of variables can vary, and can be range from 3, 4 or 300 and so on. You are required to create a function that takes the URL and those variables and returns a new update URL with the given variables' values placed at appropriate spots. The problem is to be solve using Kotlin Programming Language.

Hint: Reflection in Kotlin.

Constraints:

- Function should accept variable number of arguments.
- Function signature should not change.
- Only till functions, Use of Classes not allowed.

Sample Code:

```
fun main() {
val url = "play.kotlin.org/?id={id}/program={program}/device={device}"
val id = 1
val program = "game"
val device = "ios"
val newUrl = url.buildUrl(id,program,device)
```

```
//"play.kotlin.org/?id=1/program=game/device=ios"

print(newUrl)

}
```

Solution:

Brainstorming:

- As the number of that are passed in the function can be variable so we can use *vararg* in Kotlin to enable the function to take variable number of arguments.
- Also, the hint suggests that the problem can be solved using **Reflection in Kotlin,** will need to research on that as well**.**
- Other than that, another solution that comes to my mind is that we can iterate over all the passed arguments and search and replace the passed values in the URL using .replace() function.
- Another way is to identify the placeholders using Regex and use .replace() function.
- I can also manually search for starting { and ending } to identify placeholders and use .replace() function.

Assumptions:

- Assumed that the URL pattern does not change and placeholders are represented by {}.
- Assumed that the parameters passed are always in order according to the placeholders.

Reflection in Kotlin:

Reflection in Kotlin allows us to inspect and manipulate the properties and methods of objects at runtime. It also enables us to dynamically access and invoke elements of a class as well.

- To include reflection in Kotlin we can use “import kotlin.reflect.*”
- To get a reference of a class we can use ClassName::class.
- To get a reference of a specific property we can use ClassName::propertyName.
- To get a reference of a function we can use ClassName::myFunction.
- We can access the value of a property using the “get” method.
- And we can call a function using the “call” method.

Design Decision.

It seems that we have 3 ways in which we can solve this problem. One is to iterate through all the passed arguments and search and replace them in the URL using .replace() function, other

is to use Reflection in Kotlin to solve this problem and the third is to use Regex to identify placeholders and use .replace() function to insert values.

- One downside of using reflection can be that it may be slow and resource-intensive as reflection operations requires additional processing and memory overhead.
- Other than that, the scope of the problem is not much large so simply solving using the .replace() function seems to be a much more feasible option.
- Also, considering that this function is to be deployed in an app so adding additional libraries to solve small problems, where a solution without using any additional libraries exists can lead to a declined performance of the application.
- Also using reflection would require to use a data class to get the argument name, why (I have discussed in the Problems Encountered Section).

Now it seems to me that using .replace() function instead of Reflection is a much more efficient and feasible solution to the given problem. But we will need to still identify placeholders in the given URL. To do that we can either use Regex or manually search for placeholders using prefix and suffix as '{' and '}'. As regex seems as an advance topic given the scope of this assignment so I have decided to use .startIndex() to search for the starting and ending index of '{' and '}', and then extract the substring between the parenthesis and use .replace() function to insert the appropriate values.

Problems Encountered:

1. One problem that I faced during solving this Assignment is that after passing the variables as arguments in my function using *vararg* is that now in the function I have values of the passed arguments rather than the names of the arguments so if I use the .replace() function, I am basically searching for "1" in the URL rather than "id" in the URL.
    - A solution to this is to either use a map or to pass the arguments as a pair but that would change the function signature so might need to opt a solution using Reflection in Kotlin.
2. Another problem is that we can identify the name of parameter using reflection if we have fixed number of arguments but in case of vararg getting the reference of function using val ref = ::functionName and then use ref.parameters to get the parameters does not give away the name of parameters.
3. Also, in case of fixed number of arguments in a function the name that are returned as parameter names are actually the names of the parameters that are declared in the function declaration i.e. myFun(val a : Any, val b: Any) with function call myFun(x,y) returns a and b as parameter names which is actually wrong.