Exercise – Spring Container

Naelah Alshibani

Q1. the output:

"hey from message1"

Because there is only one method in the class (getMessage1()) and this method doesn't require any parameter, therefore it doesn't have any dependencies and can just print the message and store return value "1" in the container.

Q2. the output:

"hey from message1"

"hey from message2"

The reason why the first method got executed first is because it doesn't have any parameters and therefore doesn't have any dependencies and can be executed first, and since this method has qualifier 1, which is mandatory for method2 to execute because it takes String type as parameter, this qualifier will make the return value of getMessage1 be sent as a parameter for message2 because it has the same qualifier in its parameter, and then after it get the necessary string value method getMessage2 will execute and print "hey from message2" and then store data variable in the container.

Q3.

-the first output:

"hey from message1"

"hey from message3"

"hey from message2"

The first output will excute the getMessage1 method first because it doesn't have parameter and therefore doesn't have any dependencies and it will print "hey from message1" and then store "1" in the container, the second method that will be executed is getMessage3 for the same reason, because it doesn't have parameters and therefore doesn't have any dependencies and it will print "hey from message3" and store "3" in the container. And since this method has qualifier 3 which used in getMessage2 method parameter, it will send the return value "3" to getMessage2 that will take as a data parameter because it has the same qualifier 2, and then it will print "hey from message2".

-the second output:

“hey from message3”

“hey from message1”

“hey from message2”

This case will print the getMessage3 first because it doesn’t have parameter and therefore doesn’t have any dependencies and it will store return value “3” in the container, which later will be used for method getMessage2 because it has the same qualifier as parameter, and in this case this makes method getMessage1 and getMessage2 equals in the order of implementation, in this output case it will print getMessage1 because it doesn’t have dependencies and it will print “hello from message1”, and then it will execute getMessage2 which is ready because it has the return value from message3 qualifier and it will print “hey from message2”

-the third output:

“hey from message3”

“hey from message2”

“hey from message1”

This case will print getMessage3 because it doesn’t have any dependencies and because it has qualifier 3 it will send the return value “3” to execute getMessage2 that takes the same qualifier as parameter, and finally it will print getMessage1.

Q4.

-the first output:

“hey from message1”

“hey from main controller”

“hey from message3”

“hey from message2”

This output case will execute getMessage1 first because it doesn’t have parameters and therefore doesn’t have any dependencies, and since this method has qualifier 1 it will send the return value “1” to the MainController constructor because it takes the same qualifier before its string parameter and then it will print the message. After that it will execute getMessage3 because it doesn’t have any dependencies and since it has qualifier 3 it will send the return value to getMessage2 parameter because it has the same qualifier and it will be executed lastly after getMessage3.

-the second output:

“hey from message1”

“hey from message3”

“hey from main controller”

“hey from message2”

This output will execute getMessage1 because it doesn’t have any dependencies and because it has qualifier 1 it will send the return value to MainController constructor, this case will make getMessage3 and the constructor on equal terms of implementation order, in this case it will print getMessage3 first because it doesn’t have any dependencies and then it will execute the MainController because its parameter value is ready from qualifier 1, and then lastly it will execute getMessage2 that got its parameter value from the same qualifier of method 3 return value.

-the third output:

“hey from message3”

“hey from message1”

“hey from main controller”

“hey from message2”

This output will print “hey from message3” because getMessage3 dosent have any parameters and therefore doesn’t have any dependencies, and then it will execute getMessage1 for the same reason because it doesn’t have any dependencies, and since getMessage1 has qualifier1 it will send the return value to MainController constructor because it has the same qualifier for its parameter and then it will be executed, and since getMessage3 got executed first and it has qualifier3 that is mandatory for getMessage2 to be executed because it takes String parameter and has the same qualifier3 now it will execute getMessage2.

-the 4th output:

“hey from message3”

“hey from message2”

“hey from message1”

“hey from main controller”

This output will execute getMessage3 because it doesn't have any dependencies and since it has qualifier 3 that is mandatory for getMessage2 it will send the return value to getMessage2, this case will make getMessage1 and getMessage2 on equal terms of order execution, in this case it will implement getMessage2, and then it will implement getMessage1 because it doesn't have dependencies and since it has qualifier1 it will send the return value to mainController constructor and the constructor will be executed lastly because it has the same qualifier as getMessage1.

Q5.

-the output:

"hey from message3"

"hey from message2"

"hey from main controller"

"hey from message1"

The first method that will be executed is getMessage3 because it's the only method that doesn't have any parameters and therefore doesn't have any dependencies, and since this method has qualifier3, which is the same qualifier for getMessage2 parameter this will make getMessage2 the second method to be executed and it will get the return value from getMessage3 and then will complete execution and print "hey from message2", now since getMessage2 has qualifier2 which is the same qualifier for MainController parameter, the constructor will be the third in execution, and the last method to get executed is getMessage1 it doesn't have the same qualifier as the constructor but since it takes mainController parameter and the constructor implicitly is inside the container, this will make getMessage1 execution always dependent on MainController execution, without the need for qualifier because it's the only method that takes mainContoller as parameter and finally "hey from message1" will be printed.