Exercise (Spring Container)

Q1 output:

Concole:

hey from message1

Explanation: the message (hey from message1) will be printed in the console and the string return value ("1") will be saved in the Spring Container.

Q2 output:

Concole:

hey from message1

hey from message2

Explanation: the execution of *getMessage2* method depends on the method *getMessage1 because of using the annotation @Qualifier("1") in its parameter*. So firstly, the message (hey from message1) will be printed in the console and the string return value ("1") will be saved in the Spring Container. Secondly, the message (hey from message2) will be printed in the console and the string return value data which is the string return value of *getMessage1 method, and its value is* ("1") will be saved in the Spring Container.

Q3 output:

Concole:

 # Possibility 1

hey from message1

hey from message3

hey from message2

 # Possibility 2

hey from message3

hey from message2

hey from message1

 # Possibility 3

hey from message3

hey from message1

hey from message2

Explanation: the execution of *getMessage2* method depends on the method *getMessage3 because of using the annotation @Qualifier("3") in its parameter*. But both *getMessage1 and getMessage3 are equal in priority*.

First possibility, the message (hey from message1) will be printed in the console and the string return value ("1") will be saved in the Spring Container. Then, the message (hey from message3) will be printed in the console and the string return value ("3") will be saved in the Spring Container. Lastly, the message (hey from message2) will be printed in the console and the string return value data which is the string return value of *getMessage3 method, and its value is* ("3") will be saved in the Spring Container.

Second possibility, the message (hey from message3) will be printed in the console and the string return value ("3") will be saved in the Spring Container (now both getMessage1 and getMessage2 have the same priority). Then, the message (hey from message2) will be printed in the console and the string return value data which is the string return value of *getMessage3 method, and its value is* ("3") will be saved in the Spring Container.

Lastly, the message (hey from message1) will be printed in the console and the string return value ("1") will be saved in the Spring Container.

Third possibility, the message (hey from message3) will be printed in the console and the string return value ("3") will be saved in the Spring Container (now both getMessage1 and getMessage2 have the same priority). Then, the message (hey from message1) will be printed in the console and the string return value ("1") will be saved in the Spring Container. Lastly, the message (hey from message2) will be printed in the console and the string return value data which is the string return value of *getMessage3 method, and its value is* ("3") will be saved in the Spring Container.

Q4 output:

Concole:

 # Possibility 1

hey from message1

hey from Main controller

hey from message3

hey from message2

 # Possibility 2

hey from message1

hey from message3

hey from message2

hey from Main controller

 # Possibility 3

hey from message1

hey from message3

hey from Main controller

hey from message2

 # Possibility 4

hey from message3

hey from message2

hey from message1

hey from Main controller

 # Possibility 5

hey from message3

hey from message1

hey from message2

hey from Main controller

 # Possibility 6

hey from message3

hey from message1

hey from Main controller

hey from message2

Explanation: **the method getMessage2 depends on getMessage3 due to the use of the @Qualifier("3") annotation in its parameter. This requires that getMessage3 executes before getMessage2. Also, the MainController constructor depends on getMessage1 due to the use of the @Qualifier("1") annotation in its parameter. This requires that getMessage1 executes before MainController. However, getMessage1 and getMessage3 do not depend on each other, meaning they can be initialized in any order and have equal priority.**

First possibility, getMessage1 executes first, printing "hey from message1" to the console and storing the return value "1" in the Spring Container. Next, MainController executes, printing "hey from Main controller" and storing its data. Next, getMessage3 is executed, printing "hey from message3" and storing "3". Finally, getMessage2 executes, printing "hey from message2" and storing the return value from getMessage3, which is "3".

Second possibility, getMessage1 executes first, printing "hey from message1" and saving "1". Next, getMessage3 executes, printing "hey from message3" and storing "3". Afterward, getMessage2 executes, printing "hey from message2" and storing the data parameter value, which is "3". Finally, MainController executes, printing "hey from Main controller".

Third possibility, getMessage1 executes first, printing "hey from message1" and storing "1". Next, getMessage3 executes, printing "hey from message3" and storing "3". Next, MainController executes, printing "hey from Main controller". Finally, getMessage2 executes, printing "hey from message2" and storing "3".

Fourth possibility, getMessage3 executes, printing "hey from message3" and storing "3". Now both getMessage1 and getMessage2 are ready to execute. Next, getMessage2 executes, printing "hey from message2" and storing "3". Next, getMessage1 executes first, printing "hey from message1" and saving "1". Finally, MainController executes, printing "hey from Main controller".

Fifth possibility, getMessage3 executes first, printing "hey from message3" and storing "3". Now both getMessage1 and getMessage2 are ready to execute. Next, getMessage1 executes, printing "hey from message1" and saving "1". getMessage2 executes next, printing "hey from message2" and storing "3". Finally, MainController executes, printing "hey from Main controller".

Sixth possibility, getMessage3 executes first, printing "hey from message3" and saving "3". Then, getMessage1 executes, printing "hey from message1" and saving "1". Next, MainController executes, printing "hey from Main controller". Finally, getMessage2 executes, printing "hey from message2" and storing "3".

Q5 output:

Concole:

hey from message3

hey from message2

hey from Main controller

hey from message1

Explanation: **the only method that does not depend on another method is *getMessage3. So, because of that it will be only one possible output. Firstly,* the message (hey from message3) will be printed in the console and the string return value (“3”) will be saved in the Spring Container. After the exaction of *getMessage3 method now it’s possible to execute getMessage2 method because of using the annotation @Qualifier(“3”) in its parameter.* So secondly, the message (hey from message2) will be printed in the console and the string return value data which is the string return value of *getMessage3 method, and its value is* (”3”) will be saved in the Spring Container. Now it’s possible to execute the constructor MainController *because of using the annotation @Qualifier(“2”) in its parameter*. So thirdly, the message (hey from Main controller) will be printed in the console. Now it’s possible to execute the *execute getMessage1 because of holding a MainController object in its parameter*. So lastly, the message (hey from message1) will be printed in the console and the string return value (“1”) will be saved in the Spring Container**
