Exercise Spring Container

Abdulaziz Jaber Alshehri

Q1:

```
1 usage
@SpringBootApplication
public class SpringPollApplication {

    public static void main(String[] args) { SpringApplication.run(SpringPollApplication.class, args); }


    @Bean
    public String getMessage1(){
        System.out.println("hey from message1");
        return "1";
    }
}
```

- In this question there is one possibility output and it's:
    - hey from getMessage1.
        - Description: it's one possibility because there is one method that will be store in the container.

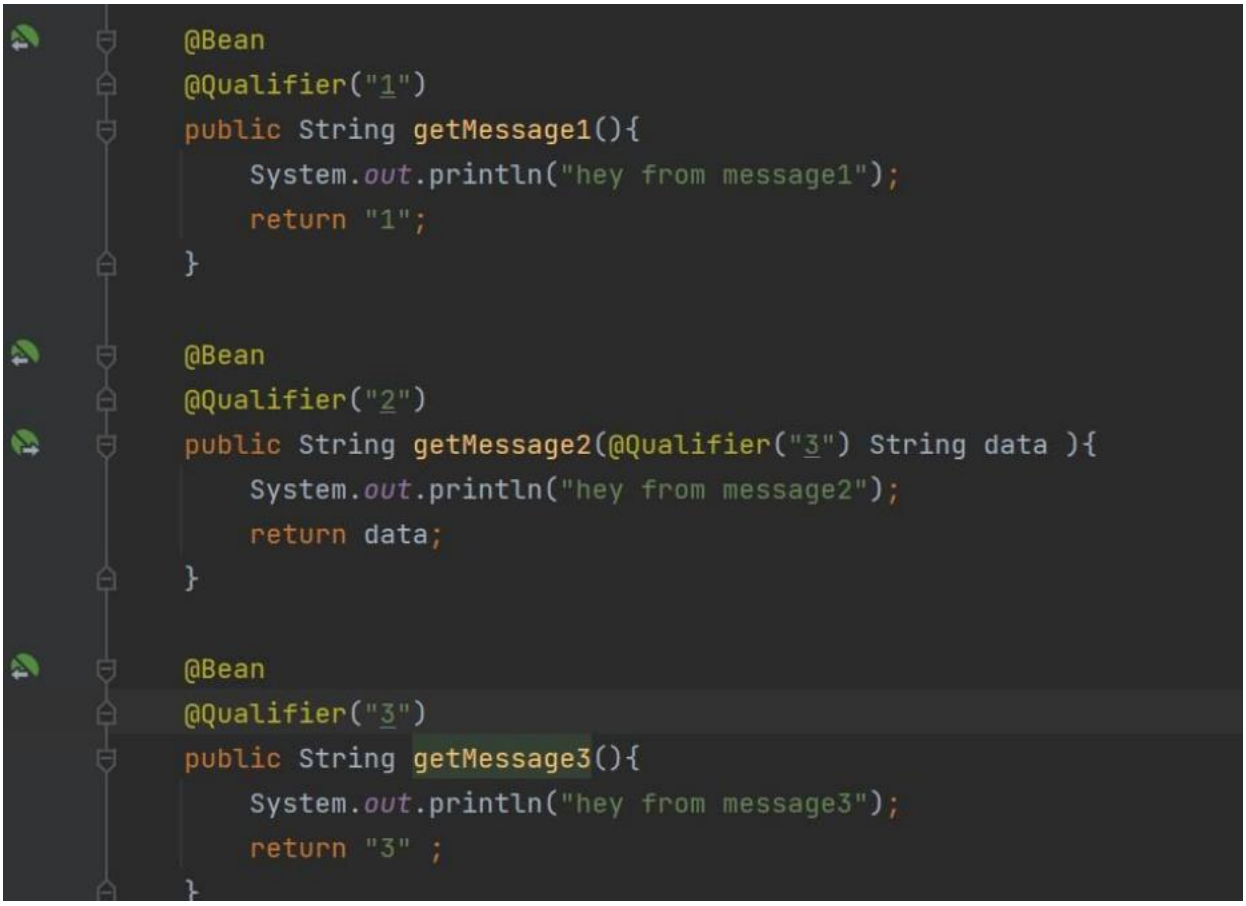
Q2:

```
1 usage
@SpringBootApplication
public class SpringPollApplication {

    public static void main(String[] args) { SpringApplication.run(SpringPollApplication.class, args); }


    @Bean
    @Qualifier("1")
    public String getMessage1(){
        System.out.println("hey from message1");
        return "1";
    }
    @Bean
    public String getMessage2(@Qualifier("1") String data ){
        System.out.println("hey from message2");
        return data ;
    }
```

- In this question there is one possibility for the output and it's:
    - hey from getMessage1.
    - hey from getMessage2.
        - Description: it's one possibility because there is a qualifier in one method and the second method doesn't have a qualifier, so obviously the first method will store in the container faster than the second method

cause the second method need a qualifier to be stored and it will get the qualifier from first method.

Q3:



```
@Bean
@Qualifier("1")
public String getMessage1(){
    System.out.println("hey from message1");
    return "1";
}

@Bean
@Qualifier("2")
public String getMessage2(@Qualifier("3") String data ){
    System.out.println("hey from message2");
    return data;
}

@Bean
@Qualifier("3")
public String getMessage3(){
    System.out.println("hey from message3");
    return "3" ;
}
```

- In this question there is two possibility for the output and it's:
    - First possibility:
    - hey from getMessage1.

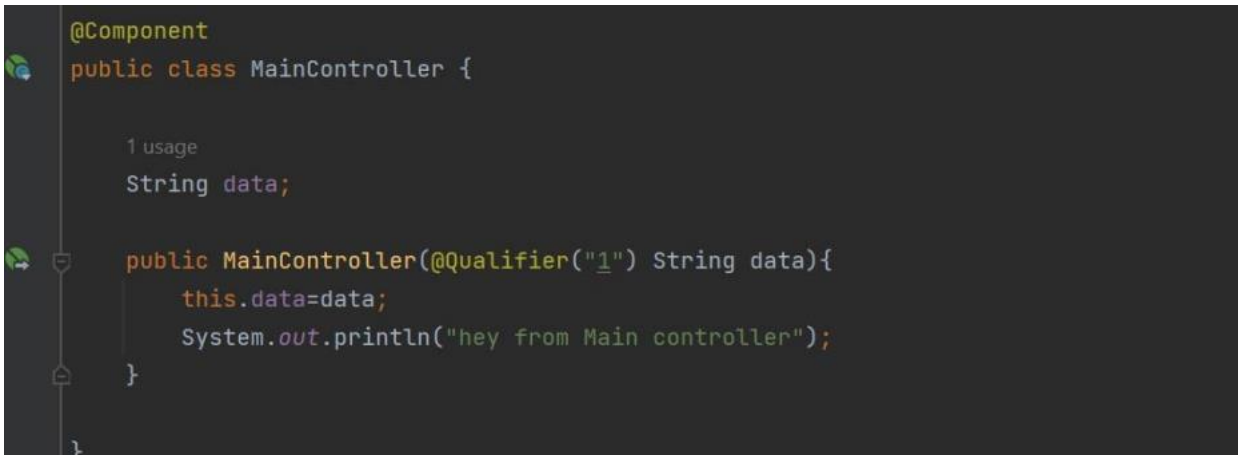
- hey from getMessage3.
- Hey from getmessage2

- Second possibility:

- hey from getMessage3.
- hey from getMessage1.
- Hey from getmessage2
    - Description: it’s two possibility because method’1’ and method’3’ doesn’t have qualifier in the parameter to be stored in the container neither like method’2’ it need a qualifier in it’s parameter to be stored in the container, so in first “possibility” getmessage’1’ will be store in the container first and getmessage’3’ will stored in second. and in the second “possibility” the getmessage’3’ will be store first then the getmessage’1’ will store second, in both possibilities getmessage’2’ is the last store

Q4:

```java
@Bean
@Qualifier("1")
public String getMessage1(){
    System.out.println("hey from message1");
    return "1";
}

@Bean
@Qualifier("2")
public String getMessage2(@Qualifier("3") String data ){
    System.out.println("hey from message2");
    return data;
}

@Bean
@Qualifier("3")
public String getMessage3(){
    System.out.println("hey from message3");
    return "3" ;
}
```



```
@Component
public class MainController {

    1 usage
    String data;

    public MainController(@Qualifier("1") String data){
        this.data=data;
        System.out.println("hey from Main controller");
    }

}
```

- In this question there is four possibility for the output and it's:
    - First possibility:
    - hey from getMessage1.
    - Hey from the main controller.
    - hey from getMessage3.
    - Hey from getmessage2.

    - Second possibility:

    - hey from getMessage3.
    - hey from getMessage1.
    - Hey from the main controller.
    - Hey from getmessage2.

    - Third possibility:

    - hey from getMessage3.
    - hey from getMessage1.
    - Hey from getmessage2.
    - Hey from the main controller.

    - Forth possibility:

- hey from getMessage1.
- hey from getMessage3.
- Hey from getmessage2.
- Hey from the main controller.
    - Description: The order of storing the data in the container is random but in the same time it's depend on the qualified method to be stored first because this methods are already have it's own data, after that the methods that need qualify in it's parameter it will stored in the container.

Q5:

```
@Bean
@Qualifier("1")
public String getMessage1(MainController mainController){
    System.out.println("hey from message1");
    return "1";
}

@Bean
@Qualifier("2")
public String getMessage2(@Qualifier("3") String data ){
    System.out.println("hey from message2");
    return data;
}

@Bean
@Qualifier("3")
public String getMessage3(){
    System.out.println("hey from message3");
    return "3" ;
}
```

-
```java
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Component;

1 usage
@Component
public class MainController {

    1 usage
    String data;

    public MainController(@Qualifier("2") String data){
        this.data=data;
        System.out.println("hey from Main controller");
    }

}
```
- In this question there is four possibility for the output and it's:
    - First possibility:
    - hey from getMessage3.
    - hey from getMessage2.
    - Hey from the main controller.
    - hey from getMessage1.

    - Second possibility:

    - hey from getMessage3.
    - Hey from the main controller.
    - Hey from getmessage2.
    - hey from getMessage1.

    - Third possibility:

- hey from getMessage1.
- hey from getMessage3.
- Hey from getmessage2.
- Hey from the main controller.

- Forth possibility:

- hey from getMessage1.
- hey from getMessage3.
- Hey from the main controller.
- Hey from getmessage2.

    - Description: “getmessage2” and “Hey from the main controller” are stored after the store of “getmessage3” because there parameter qualified by “getmessage3”.