FPR20-Java-FA2-Mock-Online

00:44:13

Reference

Finish Test

INSTRUCTIONS

SECTIONS

Expand/Collapse Sections

Simple - FPR20-Java-FA2-Mock-Online	8	Attempt Section
Medium - FPR20-Java-FA2-Mock-Online	4	Attempt Section
Complex - FPR20-Java-FA2-Mock-Online	1	Attempt Section

Scanned with CamScanner

[1 Mark]

Consider the following interface:

```
public interface CustomerRepository {
}
```

Which of the following Spring Data repositories CustomerRepository can extend to be used as a repository?

(Choose all correct optioins)

- [ ] CrudRepository
- [ ] PagingAndSortingRepository
- [ ] JpaRepository
- [ ] Repository

Reset Save

Scanned with CamScanner

[1 Mark]

Consider the following Customer entity class and CUSTOMER table with customer_id as primary key:

```
@Entity
public class Customer{
    @Id
    private Integer customerId;
    private String customerName;
    //getter and setter methods
}
```

CUSTOMER (customer_id is the primary key)

customer_id	customer_name
1	Tim
2	Chris
3	Tom
4	Jack

Assumptions:

1. All necessary imports and configurations are done

Scanned with CamScanner

Assumptions:

1. All necessary imports and configurations are done

2. transactions are managed by Spring

3. entityManager is valid reference of JPA EntityManager

What will be the output when the following code is executed?

```
Query query = entityManager.createQuery("SELECT LOWER(c.customerName) FROM Customer c where c.customerName like 'T%'")
List<String> names = query.getResultList();
System.out.println(names.get(0)+names.get(1));
```

- timtom
- TIMTom
- TimTom
- TimTOM

Reset Save

Scanned with CamScanner

Question 3

[1 mark]

Consider the following Spring beans and configuration classes:

```
package com.infy.bean;
@Component
public class Trainee {
    public Trainee() {
        System.out.print("Trainee constructor called ");
    }
}

package com.infy.bean;
@Component
public class Educator {
    public Educator() {
        System.out.print("Educator constructor called ");
    }
}

package com.infy.configuration;
```

Scanned with CamScanner

```
package com.infy.configuration;
@Configuration
@ComponentScan
public class SpringConfig {
}
```

Assumption:

1. All necessary imports and configurations are done

What will be the output when following code is executed?

```
package com.infy;
public class UserInterface {
public static void main(String[] args) throws Exception {
        ApplicationContext applicationContext = new AnnotationConfigApplicationContext(SpringConfig.class)
    }
}
```

- Educator constructor called Trainee constructor called
- Trainee constructor called Educator constructor called
- Nothing will be displayed on the console

Scanned with CamScanner

What will be the output when following code is executed?

```
package com.infy;
public class UserInterface {
public static void main(String[] args) throws Exception {
        ApplicationContext applicationContext = new AnnotationConfigApplic
    }
}
```

- Educator constructor called Trainee constructor called
- Trainee constructor called Educator constructor called
- Nothing will be displayed on the console
- NullPointerException will be thrown

Reset Save

Scanned with CamScanner

Question 4

[1 Mark]

Consider the following class:

```
@RestControllerAdvice
public class ExceptionHandler{

    @ExceptionHandler
    public ResponseEntity<String> ExceptionHandler(RunTimeException exception){
        //rest of the code
    }

}

public class CustomerException extends RuntimeException{
//rest of the code
}
```

Assumption:

1. All necessary imports and configurations are done

Scanned with CamScanner

```
}

public class CustomerException extends RuntimeException{
//rest of the code
}
```

Assumption:

1. All necessary imports and configurations are done

Which of the following statements are CORRECT about the above ExceptionHandler class?

- [ ] It will handle NullPointerException
- [ ] It will handle CustomerException
- [ ] It will handle IOException
- [ ] It will not handle any exception because exception type Is not mentioned with @ExceptionHandler annotatic

Reset Save

Scanned with CamScanner

Question 5

[1 mark]

Consider the following REST controller:

```
@RestController
@RequestMapping("/customer")
public class CustomerLoginController{
   public void login(@RequestParam("password") String pwd,@RequestParam("loginName") String name){
      System.out.print(name);
      System.out.print(pwd);
   }
}
```

Assumption:

1. All necessary imports and configurations are done

What will be the output of the above code, when a request comes with a URI **/customer?loginName=tom&p**

- ◯ tomtom456

Scanned with CamScanner

```
        System.out.print(pwd);
    }
}
```

Assumption:

1. All necessary imports and configurations are done

What will be the output of the above code, when a request comes with a URI **/customer?loginName=tom&password=tom456** ?

- tomtom456
- tom456tom
- tomnull
- nulltom

Reset Save

Scanned with CamScanner

Question 6

[1 Mark]

Consider the following entity class and corresponding table:

```
@Entity
public class Laptop{
        @Id
        @GeneratedValue(strategy = GenerationType.IDENTITY)
        private Integer laptopId;
        private String model;
        // getters and setters
}
```

LAPTOP (laptop_id is the primary key and auto-increment column)

laptop_id	model
1	probook

Assumptions:

1. All necessary imports and configurations are done

Scanned with CamScanner

LAPTOP (laptop_id is the primary key and auto-increment column)

laptop_id	*model*
1	*probook*

Assumptions:

1. All necessary imports and configurations are done
2. laptopRepository is a valid reference of LaptopRepository interface which extends CrudRepository
3. Transactions are managed by Spring

Which of the following statements are TRUE when following code is executed?

```
Laptop l = new Laptop();
l.setModel("hp305");
laptopRepository.save(l);
```

- It will insert a row in LAPTOP table with laptop_id as 2 and model as hp305
- It will insert a row in LAPTOP table with laptop_id as NULL and model as hp305

Scanned with CamScanner

2. laptopRepository is a valid reference of LaptopRepository interface which extends C

3. Transactions are managed by Spring

Which of the following statements are TRUE when following code is executed?

```
Laptop l = new Laptop();
l.setModel("hp305");
laptopRepository.save(l);
```

- It will insert a row in LAPTOP table with laptop_id as 2 and model as hp305
- It will insert a row in LAPTOP table with laptop_id as NULL and model as hp305
- The code will throw NullPointerException as laptopId is not set
- The code will throw IllegalStateException as laptopId is not set

Reset Save

Scanned with CamScanner

Question 7

[1 Mark]

Consider the following Spring bean classes:

```
package com.infy.service;
@Service
public class CustomerLoginServiceImpl {
    @Autowired
    private CustomerLoginRepositoryImpl customerLoginRepositoryImpl;
    public CustomerLoginServiceImpl(){
        System.out.println("CustomerLoginServiceImpl constructor called");
    }
    //getters & setters
}
```

```
package com.infy.repository;
@Repository
public class CustomerLoginRepositoryImpl {
    public CustomerLoginRepositoryImpl (){
        System.out.println("CustomerLoginRepositoryImpl constructor called")
    }
```

Scanned with CamScanner

```
package com.infy.repository;
@Repository
public class CustomerLoginRepositoryImpl {
    public CustomerLoginRepositoryImpl (){
            System.out.println("CustomerLoginRepositoryImpl constructor called")
    }
    //getters & setters
}
```

Assumption:

1. All the necessary imports and configurations are done

What will be the output when the following code is executed?

```
package com.infy;
@SpringBootApplication
public class Application{

    @Autowired
    CustomerLoginServiceImpl customerLoginServiceImpl;

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

Scanned with CamScanner

Assumption:

1. All the necessary imports and configurations are done

What will be the output when the following code is executed?

```
package com.infy;
@SpringBootApplication
public class Application{

    @Autowired
    CustomerLoginServiceImpl customerLoginServiceImpl;

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

- CustomerLoginServiceImpl constructor called
- CustomerLoginRepositoryImpl constructor called

Scanned with CamScanner

```
        SpringApplication.run(Application.class, args);
    }
}
```

- CustomerLoginServiceImpl constructor called
  CustomerLoginRepositoryImpl constructor called
- CustomerLoginRepositoryImpl constructor called
  CustomerLoginServiceImpl constructor called
- CustomerLoginServiceImpl constructor called
- CustomerLoginRepositoryImpl constructor called

Reset Save

Scanned with CamScanner

Question 8

[1 Mark]

Which of the following annotation is used to bind a JPQL query with a method in Spring Data?

- @Query
- @NamedQuery
- @PersistenceContext
- None of these

Reset Save

Scanned with CamScanner

Medium - FPR20-Java-FA2-Mock-Online

1 2

3 4

Question 1

[2 marks]

Consider the following class:

```
package com.infy.utility;
@Component
@Aspect
public class LoggingAspect {
    @Around("execution(* com.infy.Application.*(..))")
    public void aroundAdvice(ProceedJoinPoint pjp) throws Exception {
        System.out.println("In around advice");
    }
}
```

Assumption:

1. All necessary imports and configurations are done

What will be the output when the following code is executed?

```
package com.infy;
@SpringBootApplication
```

Scanned with CamScanner

Assumption:

1. All necessary imports and configurations are done

What will be the output when the following code is executed?

```
package com.infy;
@SpringBootApplication
public class Application implements CommandLineRunner {
    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }
    public void run(String... args) throws Exception {
        display();
    }
    public void display() throws Exception {
        System.out.println("display method called");
    }
}
```

○ In around advice

Scanned with CamScanner

```
    public void display() throws Exception {
        System.out.println("display method called");
    }
}
```

- In around advice
- In around advice
  display method called
- display method called
- NullPointerException will be thrown

Reset Save

Scanned with CamScanner

Question 2

[2 Marks]

Consider the following entity class which is mapped to LOCKER table with no rows

```
@Entity
public class Locker {
    @Id
    private String lockerId;
    private String lockerType;
    //getters & setters
}
```

Assumptions:

1. All necessary imports and configuration are done
2. entityManager is a valid reference of JPA EntityManager
3. Transaction is managed by Spring

Which of the following option is TRUE about the below code?

```
Locker locker=new Locker();
locker.setLockerId("LOC10001");
```

Scanned with CamScanner

Which of the following option is TRUE about the below code?

```
Locker locker=new Locker();
locker.setLockerId("LOC10001");
locker.setLockerType("Small");
entityManager.persist(locker);
entityManager.persist(locker);
locker = entityManager.find(Locker.class, "LOC10001");
System.out.println(locker.getLockerType());
```

- Code will throw UniqueConstraintViolationException exception
- Small
- Code will throw NullPointerException exception
- Code will throw SQLIntegrityConstraintException exception

Reset Save

Scanned with CamScanner

Question 3

[2 marks]

Consider the following URI:

```
String url = "http://www.xyz.com/customers/{customerId}";
```

Which of the following is correct for use of RestTemplate for sending GET request to above url to fetch details of customer whose customerId = 1001 and is represented by Customer model class?

- ○
```
RestTemplate restTemplate = new RestTemplate();
restTemplate.getForObject(url, Customer.class,1001);
```

- ○
```
RestTemplate restTemplate = new RestTemplate();
restTemplate.get(url, Customer.class,1001);
```

- ○
```
RestTemplate restTemplate = new RestTemplate();
restTemplate.getObject(url, Customer.class,1001);
```

Scanned with CamScanner

Which of the following is correct for use of RestTemplate for sending GET request to customerId = 1001 and is represented by Customer model class?

- ```
  RestTemplate restTemplate = new RestTemplate();
  restTemplate.getForObject(url, Customer.class,1001);
  ```
- ```
  RestTemplate restTemplate = new RestTemplate();
  restTemplate.get(url, Customer.class,1001);
  ```
- ```
  RestTemplate restTemplate = new RestTemplate();
  restTemplate.getObject(url, Customer.class,1001);
  ```
- None of these

Reset Save

Scanned with CamScanner
  ```

[2 marks]

Consider the below table:

```
create table loan(
loan_id int primary key,
loan_type varchar(20));
```

Assumptions:

1. all necessary imports and configurations are done
2. SpringPhysicalNaming strategy is followed

Which of the following entity class declarations is/are properly mapped to above loan table?

i.

```
@Entity
public class Loan {
    @Id
    private Integer loanId;
    private String loanType;
    //getters & setters
}
```

Scanned with CamScanner

ii.

```
@Entity
@Table(name="LOAN")
public class Laon {
    @Id
    private Integer loan_id;
    private String loan_type;
    //getters & setters
}
```

- Both i and ii
- Only i
- Neither i nor ii
- Only ii

Resel Save

Scanned with CamScanner

Complex - FPR20-Java-FA2-Mock-Online

1

Question 1

[3 Marks]

Consider the following tables:

Customer (customer_id is primary key)

customer_id	customer_name
101	Mona

Locker (locker_id is primary key and customer_id is foreign key referencing customer_id of Customer table)

locker_id	locker_type	customer_id
L201	Small	101
L202	Medium	101

Customer and Locker entity classes are mapped to Customer and Locker tables respectively and there is unidirectional one-to-many mapping from Customer to Locker.

Assumptions:

1. All necessary imports and configuration are done

Scanned with CamScanner

Customer and Locker entity classes are mapped to Customer and Locker tables respectively and there is unidirectional one-to-many mapping from Customer to Locker.

Assumptions:

1. All necessary imports and configuration are done

2. transactions are managed by Spring

3. lockerRepository is a valid reference of LockerRepository interface which extends CrudRepository

Which of the following option is correct when the below code is executed?

```
lockerRepository.deleteById("L202");
```

- ○ No records will be deleted from either of the tables
- ○ In Locker table L202 row will be deleted successfully
- ○ In Locker table L202 row and in Customer table 101 row will be deleted successfully
- ○ In Locker table L201 and L202 rows and in Customer table 101 row will be deleted successfully

Scanned with CamScanner