As:

Во кој слој прикаѓаат класите од Spring MVC анотирани со @Controller?

Select one:

a. Presentation

b. Persistence

c. User Interface

d. Domain Model

e. Service

Доколку имаме потреба да пристапиме до сесиски атрибут "user" во рамките на мапиран метод во Spring MVC, во кој од следните методи е можно да го пристапиме атрибутот?

Select one or more:

a.

```
@Controller
@RequestMapping("/exam")
public class ExamController {

 @RequestMapping("question")
 public String question(HttpServletRequest request) {
   ...
```

b.

```
@Controller
@RequestMapping("/exam")
public class ExamController {

 @RequestMapping("question")
 public String question(ServletContext context) {
   ...
```

c.

```
@Controller
@RequestMapping("/exam")
public class ExamController {

 @RequestMapping("question")
 public String question(HttpSession session, ServletContext context) {
   ...
```

d.

```
@Controller
@RequestMapping("/exam")
public class ExamController {

 @RequestMapping("question")
 public String question(@Session HttpSession session) {
   ...
```

e.

```
@Controller
@RequestMapping("/exam")
public class ExamController {

 @RequestMapping("question")
 public String question(@SessionAttribute(name="user") Object user) {
   ...
```

На кое URL ќе треба да се направи барање за да се пристапи методот:

```
@RequestMapping(value = {"/search"}, method = RequestMethod.GET)
public String search(
```

@RequestParam String query,

Model model

) { ... }

Претпоставете дека апликацијата е стартувана на localhost:8080/

Select one:

a. GET http://localhost:8080/search/model?query=text

b. GET http://localhost:8080/search?query=text&model=

c. GET http://localhost:8080/search?query=text

d. GET http://localhost:8080/search/query

Кои искази се точни за Spring MVC.

Select one or more:

a. Spring MVC ги процесира барањата преку DispatcherServlet-от

b. Spring MVC ги процесира барањата преку DispatcherFilter-от

c. Spring MVC ги вичитува зависностите (bean) преку ContextLoaderListener од локацијата дефинирана во contextConfigLocation параметарот

d. Spring MVC ги вичитува зависностите (bean) од локацијата дефинирана во contextConfigLocation сервлет иницијализацискиот параметар

Previous page

Поврзете ги настаните од животниот циклус на ентитетите со броевите на сликата.

1

Answer 1

2

Answer 2

3

Answer 3

4

Answer 4

5

Answer 5

6

Answer 6

7

Answer 7

Кое од следните мапирања на класите Training и Group е валидно?

Select one or more:

a.

```
@Entity
public class Training {

  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  public Long id;
  public String name;
  @ManyToOne
  public Group group;

}
```

```
@Entity
public class Group {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  public Long id;
  public String name;
  @OneToMany(mappedBy = "TRAINING")
  List<Training> trainings;
}
```

b.

```
@Table(name = "TRAINING")
@Entity
public class Training {

  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  public Long id;
  public String name;
  @ManyToOne
  public Group group;

}

@Table(name = "TRAINING_GROUP")
@Entity
public class Group {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
  public Long id;
  public String name;
  @OneToMany(mappedBy = "training")
  List<Training> trainings;
}
```

c.

```
@Table(name = "TRAINING")
@Entity
public class Training {

  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  public Long id;
  public String name;
  @ManyToOne
  public Group group;

}

@Table(name = "TRAINING_GROUP")
@Entity
public class Group {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  public Long id;
  public String name;
  @OneToMany(mappedBy = "group")
  List<Training> trainings;
}
```

d.

```
@Table(name = "TRAINING")
@Entity
public class Training {

 @Id
 @GeneratedValue(strategy = GenerationType.IDENTITY)
 public Long id;
 public String name;
 @OneToMany
 public Group group;

}

@Table(name = "TRAINING_GROUP")
@Entity
public class Group {
 @Id
 @GeneratedValue(strategy = GenerationType.IDENTITY)
 public Long id;
 public String name;
 @ManyToOne(mappedBy = "group")
 List<Training> trainings;
}
```

e.

```
@Table(name = "TRAINING")
@Entity
public class Training {
```

```
 @Id
 @GeneratedValue(strategy = GenerationType.IDENTITY)
 public Long id;
 public String name;
 @ManyToOne
 public Group group;

}

@Table(name = "TRAINING_GROUP")
@Entity
public class Group {
 @Id
 @GeneratedValue(strategy = GenerationType.IDENTITY)
 public Long id;
 public String name;
 @Transient
 List<Training> trainings;
}
```

Потребно е да се напише JPA класа која ќе се поврзе со табелата COOPERATE_USER. Секој корисник е единствено идентификуван со неговиот матичен број EMBG. Покрај матичниот број за корисниците се чуваат име, работа, адреса и датум на раѓање

Како ќе го декларирате ваквиот JPA ентитет (селектирајте ги исказите)?

Select one:

a.

```
@Table(name = "COOPERATE_USER")
@Entity
```

```
public class CooperateUser {

  @Id
  public String embg;

  public String name;

  public String job;

  public String address;

  public DateTime birthDate;



}
```

b.

```
@Table(name = "COOPERATE_USER")
@Entity
public class CooperateUser {

  @IdColumn(name = "EMBG")
  public String embg;

  public String name;

  public String job;

  public String address;
```

```
  public DateTime birthDate;



}
```

c.

```
@Table(name = "COOPERATE_USER")
public class CooperateUser {

  @Id
  @Column(name = "EMBG")
  public String embg;

  public String name;

  public String job;

  public String address;

  public DateTime birthDate;



}
```

d.

```
@Entity
public class CooperateUser {

  @Id
  @Column(name = "EMBG")
  public String embg;
```

```
  public String name;

  public String job;

  public String address;

  public DateTime birthDate;

}
```

e.

```
@Table(name = "COOPERATE_USER")
@Entity
public class CooperateUser {

  @Id
  @Column(name = "EMBG")
  public String embg;

  public String name;

  public String job;

  public String address;

  public DateTime birthDate;


}
```

Доколку го имаме следното мапирање на ентитетите, кои табели ќе се изгенерираат од страна на JPA.

```
@Entity
@Table(name = "STUDENTS")
public class Student extends BaseEntity {

  @Column(name = "student_index")
  public String index;
  public String firstName;
  public  String  lastName;
  @ManyToMany
  public List<Course> courses;
}

@Entity
@Table(name = "COURSES")
public class Course extends BaseEntity {

  public String name;
  @ManyToMany(mappedBy = "courses")
  public List<Student> students;

}
```

Притоа ја имаме следната конфигурација:

```
<property name="jpaVendorAdapter">
    <bean class="org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter"
          p:database="${jpa.database}"
```

```
        p:generateDdl="true"
        p:showSql="false"/>
</property>
```

Select one or more:

a. COURSES

b. COURSES_STUDENTS

c. STUDENTS

d. STUDENT_COURSES

Даден е ентитетот:

```
@Entity
@Table(name = "book_details")
public class BookDetails extends BaseEntity {

  @Column(length = 5000)
  public String description;

  @OneToOne
  public Book book;

}
```

Доколку сакаме да ги пронајдеме сите вредности кои содржат одреден опис, кој метод треба да се додаде во следната класа подолу за Spring Data да ни овозможи добивање на валидните резултати:

```
@Repository
public interface BookDetailsRepository extends CrudRepository<BookDetails, Long> {
```

}

Select one:

a. List<BookDetails> searchDetails(String text);

b. List<BookDetails> findByDetailsLike(String text);

c. List<BookDetails> findByBookDetailsDescriptionLike(String text);

d. List<BookDetails> findByDetails(String text);

Previous page

Кој ги делегира барањата на Spring Security Filter Chain?

Select one:

a. ApplicationContext

b. DispatcherServlet

c. ContextLoaderListener

d. DelegatingFilterProxy

На линијата за одговорот подредете ги чекорите кај Test Driven Development (буквите пред секој од одговорите, одделени со запирка).

Додавање на тест 1

Стартување на тестовите одново, се додека не се осигура дека системот работи правилно44

Рефакторирање на системот за да се подобри неговиот дизајн и да се отстранат редундантните делови5

Се прават мали промени, за да поминат тестовите3

Стартување на сите тестови и проверка дека системот не

работи2

Стартување на сите тестови и проверка дека системот работи

Напомена: Не мора да се искористат сите предлози. Одговорот не смее да содржи празни места

Answer:

Кој тип на тестирање се користи за проверка дали просечното време на одговор на веб апликацијата одговарана SLA (Service Level Agreement)?

Select one:

a. Unit Testing

b. Integration Testing

c. Performance Testing ?????

d. Stress Testing

e. User Acceptance Testing

f. System Testing

WP kolokvium:

Кој од следниве парчиња код се валидни како REST сервис за добивање на податоци за сите вработени.

Забелешка: Сметајте дека соодветните мапирања и поставувања на ниво на класа и контролер се направени соодветно на потребите.

Select one or more:

a.

```
@GetMapping
private Flux<Employee> getAllEmployees() {
   return employeeRepository.findAllEmployees();
}
```

b.

```
@Bean
RouterFunction<ServerResponse> getAllEmployeesRoute() {
 return route(GET("/employees"),
   req -> ok().body(
    employeeRepository().findAllEmployees(), Employee.class));
}
```

c.

```
@Bean
RouterFunction<ServerResponse> composedRoutes() {
 return
   route(GET("/employees"),
     req -> ok().body(
       employeeRepository().findAllEmployees(), Employee.class))

   .and(route(GET("/employees/{id}"),
     req -> ok().body(
       employeeRepository().findEmployeeById(req.pathVariable("id")), Employee.class)))

   .and(route(POST("/employees/update"),
     req -> req.body(toMono(Employee.class))
       .doOnNext(employeeRepository()::updateEmployee)
       .then(ok().build())));
}
```

d.

```
@GetMapping
private Mono<Employee> getAllEmployees() {
   return employeeRepository.findAllEmployees();
}
```

Кои од следните тврдења се точни за AuthenticationProvider?

Select one or more:

a. Во иста апликација може да има конфигурирано повеќе од еден AuthenticationProvider

b. AuthenticationProvider служи за најава со Authentication објект

c. AuthenticationProvider го управува процесот на најава кај Spring Security

d. AuthenticationProvider служи за најава само со корисничко име и лозинка

Даден е ентитетот:

```
@Entity
@Table(name = "book_details")
public class BookDetails extends BaseEntity {

  @Column(length = 5000)
  public String description;

  @OneToOne
  public Book book;

}
```

Доколку сакаме да ги пронајдеме сите вредности кои имаат одреден опис, кој метод треба да се додаде во следната класа подолу за Spring Data да ни овозможи добивање на валидните резултати:

```
@Repository
public interface BookDetailsRepository extends CrudRepository<BookDetails, Long> {

}
```

Select one:

a. List<BookDetails> findByDescription(String text);

b. List<BookDetails> searchDetails(String text);

c. List<BookDetails> findByBookDetailsDescriptionLike(String text);

d. List<BookDetails> findByDescriptionLike(String text);

Доколку го имаме следното мапирање на ентитетите, кои табели ќе се изгенерираат од страна на JPA.

```
@Entity
@Table(name = "STUDENTS")
public class Student extends BaseEntity {

 @Column(name = "student_index")
 public String index;
 public String firstName;
 public  String  lastName;
 @ManyToMany
 public List<Course> courses;
```

```
}

@Entity
@Table(name = "COURSES")
public class Course extends BaseEntity {

  public String name;
  @ManyToMany
  public List<Student> students;

}
```

Притоа ја имаме следната конфигурација:

```
<property name="jpaVendorAdapter">
   <bean class="org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter"
         p:database="${jpa.database}"
         p:generateDdl="true"
         p:showSql="false"/>
</property>
```

Select one or more:

a. STUDENTS

b. STUDENTS_COURSES

c. COURSES_STUDENTS

d. COURSES

Потребно е да се дефинира нарачка за ресторан. За таа цел е дефиниран ентитетот:

```
public class Order {
  public String orderId;
  public String clientContact;
  public int persons;
  public String table;
  public DateTime date;
}
```

Кој од следните мапирања се валидни за овој ентитет?

Select one or more:

a.

```
@Table(name = "RESTAURANT_ORDER")
public class Order {

  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  public String orderId;

  public String clientContact;

  public int persons;

  public String table;

  public DateTime date;

}
```

b.

```
@Table(name = "RESTAURANT_ORDER")
```

```
@Entity
public class Order {

  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  public String orderId;

  public String clientContact;

  public int persons;

  @Column(name = "table_number")
  public String table;

  public DateTime date;

}
```

c.

```
@Entity
public class Order {

  public String orderId;

  public String clientContact;

  public int persons;

  public String table;
```

```
 public DateTime date;

}
```

d.

```
@Table(name = "RESTAURANT_ORDER")
@Entity
public class Order {

 @Id
 public String orderId;

 public String clientContact;

 public int persons;

 @Column(name = "table_number")
 public String table;

 public DateTime date;

}
```

e.

```
@Entity
public class Order {

 @Id
 public String orderId;

 public String clientContact;
```

public int persons;

public String table;

public DateTime date;

}

??????????????????

Кое од следните мапирања на класите Training и Group е валидно?

Select one or more:

a.

```
@Table(name = "TRAINING")
@Entity
public class Training {

 @Id
 @GeneratedValue(strategy = GenerationType.IDENTITY)
 public Long id;
 public String name;
 @OneToMany
 public Group group;

}

@Table(name = "TRAINING_GROUP")
```

```
@Entity
public class Group {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  public Long id;
  public String name;
  @ManyToOne(mappedBy = "group")
  List<Training> trainings;
}
```

b.

```
@Entity
public class Training {

  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  public Long id;
  public String name;
  @ManyToOne
  public Group group;

}

@Entity
public class Group {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  public Long id;
  public String name;
  @OneToMany(mappedBy = "TRAINING")
```

```
  List<Training> trainings;
}
```

c.

```
@Table(name = "TRAINING")
@Entity
public class Training {

  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  public Long id;
  public String name;
  @ManyToOne
  public Group group;

}

@Table(name = "TRAINING_GROUP")
@Entity
public class Group {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  public Long id;
  public String name;
  @OneToMany(mappedBy = "training")
  List<Training> trainings;
}
```

d.

```
@Table(name = "TRAINING")
@Entity
```

```
public class Training {

  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  public Long id;
  public String name;
  @ManyToOne
  public Group group;

}

@Table(name = "TRAINING_GROUP")
@Entity
public class Group {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  public Long id;
  public String name;
  @OneToMany(mappedBy = "group")
  List<Training> trainings;
}
```

e.

```
@Table(name = "TRAINING")
@Entity
public class Training {

  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  public Long id;
```

```
  public String name;
  @ManyToOne
  public Group group;

}

@Table(name = "TRAINING_GROUP")
@Entity
public class Group {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  public Long id;
  public String name;
  @Transient
  List<Training> trainings;
}
```

Нека во веб апликација е поставена следната конфигурацијана view-resolver:

```
<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
   <property name="prefix" value="/WEB-INF/views/"/>
   <property name="suffix" value=".jsp"/>
</bean>
```

Која ќе биде датотеката која ќе се избере од ViewResolver-от за приказ (view template) при повик на question() методот дефиниран во продолжение?

```
@Controller
@RequestMapping("/exam")
public class ExamController {
```

```
 @RequestMapping(value = "/question", method = RequestMethod.POST)
 public String question() {
  return "question";
 }
}
```

Select one:

a. /views/question.jsp

b. Нема да се селектира ниту една датотека, ќе се генерира JSON

c. Нема да се селектира ниту една датотека, ќе се изврши редирекција

d. /WEB-INF/jsp/question.jsp

e. /WEB-INF/views/question.jsp

Во кој од подолу предложените слоеви е најлогично да биде следната имплементација:

```
public List<User> getAllUsers() throws IOException {
 File usersSource = new File(SOURCE_PATH);
 BufferedReader br = new BufferedReader(new InputStreamReader(new
FileInputStream(usersSource)));
 CSVFormat format = CSVFormat.DEFAULT.withHeader().withDelimiter(',');
 CSVParser parser = new CSVParser(br, format);
 List<User> users = new ArrayList<User>();
 for (CSVRecord record : parser)
 users.add(parseUser(record));
 parser.close();
```

```
  br.close();
  return users;
}
```

Select one:

a. User Interface

b. Service

c. Persistence

d. Domain Model

e. Presentation

Question 12

Not yet answered

Marked out of 1.00

Not flaggedFlag question

Question text

Кои од следните анотации ги вклучуваат класите во Spring контекстот?

Select one or more:

a. @Bean

b. @Service

c. @Singleton

d. @Context

e. @Prototype

f. @Configuration

Во кои од следните сценарија логично е атрибутите да се чуваат во Servlet Context (application scope)?

Select one or more:

a. Подржани валути за плаќање

b. ID на тековната сесија

c. Корисничкото име при најава

d. Јазикот на корисникот

e. Резултати од пребарување

f. Контакт информации за страната

Доколку сакате да се регистрирате за добивање на пораки на одреден Publisher на податоци, во кој метод од сервлетот ќе го сторите тоа?

Select one:

a. Конструктор

b. doPost

c. service

d. init

e. doGet

f. destroy

Различни корисници од различни прелистувачи (browsers) имаат пристап до ист податок зачуван во:

Select one or more:

a. Request

b. Session

c. Query String

d. ServletContext (application context)

e. Cookie

Доколку во ServletA повикаме dispatcherToServletB.forward(req, resp), што од следното е точно:

Select one or more:

a. Во ServletB можеме да ги пристапиме Request атрибутите поставени во ServletA

b. Во ServletB можеме да ги пристапиме Session атрибутите поставени во ServletA

c. Контејнерот ќе го повика сервлетот мапиран на servletB.do

d. Во ServletB можеме да ги пристапиме параметрите пратени до ServletA

e. Клиентот ќе направи ново барање до /servletB.do

Селектирајте дали исказите опишуваат Attribute или Parameter во J2EE контејнерот.

Тип String

Answer 1

Може да се променат во сервлетите

Answer 2

Не може да се променат во сервлетите

Answer 3

Тип Object

Answer 4

Овозможуваат структурирање на кодот според MVC шаблонот

Што од следното е валидно за состојба (state) во React.JS?

Select one or more:

a. Менувањето на вредностите може да се направи директно со доделување на вредност на објектот this.state надвор од конструкторот.

b. Менувањето на вредностите може да се направи директно со користење на функцијата this.setState() надвор од конструкторот.

c. Вредностите кои веќе еднаш се поставени во state на компонента, не може да се менуваат.

d. Вредностите кои веќе еднаш се поставени во state на компонента, може да се менуваат.

e. Во рамките на конструкторот на компонентата не може да се прави доделување на вредности на објектот this.state

f. Во рамките на конструкторот на компонентата може да се прави доделување на вредности на објектот this.state

Кој од понудените опции прикажува валидна синтакса на функцијата decrement за асинхроно намалување на вредноста за 1 на бројачот во променливата counter сместена во состојбата на компонентата при клик на копчето Decrement? Почетната вредност на бројачот е 100.

render() функцијата изгледа вака:

```
render() {
   return (
      <div className="text-center">
         <label className="control-label">Counter value</label>
         <h3>{this.state.counter}</h3>
         <button className="btn btn-primary" onClick={this.decrement}>Decrement</button>
      </div>
   )
}
```

Функционалноста на компонентата е дадена во следните 2 слики:

counter value 100 decrement

По клик на копчето Decrement, потребно е да го даде следниот излез:

counter value 99 decereMENT

Select one:

a.

```
decrement => {
   this.setState((state)=> ({
      counter:state.counter-1
   }));
}
```

b.

```
decrement = (e) => {
   this.setState((state)=> ({
      counter:state.counter-1
   }));
}
```

c.

```
decrement {
   this.setState((state)=> ({
      counter:state.counter-1
   }));
}
```

d.

```
decrement() {
   this.state = {
```

```
      counter:this.state.counter-1
    }
}
```

e.

```
function decrement(e) {
    this.state.counter--;
}
```

Потребно е да се имплементира форма за додавање на нов вработен во некој систем за менаџирање на вработени во рамките на една компанија. Податоците за новиот вработен се додаваат преку соодветна форма. За секој вработен треба да знаеме име, презиме и работен стаж. Сите податоци се внесуваат преку текстуални полиња. Внесувањето на нов вработен се прави со клик на копчето Додади. Податоците за вработените се чуваат во соодветна JSON листа во состојбата на компонентата. Додавањето на нов вработен во листата, треба да биде асинхроно.

Кодот на целосната компонента EmployeeComponent изгледа вака:

```
import React, {Component} from 'react'

class EmployeeComponent extends Component {
    constructor(props) {
        super(props);
        this.state = {
            employees: [
                {
                    name: "Petre",
                    surname: "Petrov",
                    workExperience: 34
                },
```

```
        {
          name: "Petar",
          surname: "Petrov",
          workExperience: 12
        }
      ]
    }
}

handleSubmit = ?

render() {

  let employeesList = this.state.employees.map((item, index) => (
    <li>{item.name}</li>
  ));

  return (
    <div className="container-fluid">
      <ul>
        {employeesList}
      </ul>
      <form onSubmit={this.handleSubmit}>
        <div className="wrapper">
          <div className="row">
            <div className="col-md-12">
              <label className="control-label">Name:</label>
              <input name={"name"} type={"text"}
                     className="form-control"/>
```

```
                    </div>
                    <div className="col-md-12">
                      <label className="control-label">Surname:</label>
                      <input name={"surname"} type={"text"}
                             className="form-control"/>
                    </div>

                    <div className="col-md-12">
                      <button type={"submit"} className="btn btn-success">Submit</button>
                    </div>
                  </div>
                </div>
              </form>
            </div>
        )
    }
}

export default EmployeeComponent;
```

Кој од следните имплементации нуди синтаксички валиден код за функцијата handleSubmit кој е соодветен во веќе приложениот код?

Select one:

a.

```
handleSubmit(e) {
    e.preventDefault();
    const employee = {
        name: e.target.name.value,
```

```
      surname: e.target.surname.value
    }
    this.setState((state) => {
      return {
        "employees":
          [state.employees,employee]
      }
    });
};
```

b.

```
handleSubmit = (e) => {
    e.preventDefault();
    const employee = {
      name: e.target.name.value,
      surname: e.target.surname.value
    }
    this.setState((state) => {
      return {
        "employees":
          [...state.employees,employee]
      }
    });
};
```

c.

```
handleSubmit = (e) => {
    e.preventDefault();
    let employee = {
      name: e.target.name.value,
      surname: e.target.surname.value
```

```
    }
    this.setState({
          "employees":
             [...state.employees,employee]
       }
    );
};
```

d.

```
handleSubmit(e) {
    e.preventDefault();
    let employee = {
       name: e.target.name.value,
       surname: e.target.surname.value
    }
    this.setState((state) => {
       return {
          "employees":
             [...state.employees,employee]
       }
    });
};
```

Избери која од понудените опции дава валидна синтакса во ReactJS за прикажување на детали за предмет во една студиска програма. Името на компонентата е Course. Податоците за курсот се предаваат како својство на следниот начин:

```
let obj={
    id:1,courseName:"WP",
    courseFund:"2+2+1"
```

};

<Course course={obj}/>

Select one:

a.

```
import React, {Component} from 'react'


class Course extends Component {


   render() {
      return (
         <li key="{this.props.course.id}" className="list-group-item">
            <div >
               <div className="row">
                  <div className="col-md-2">
                     {this.course.id}
                  </div>
                  <div className="col-md-6">
                     {this.course.courseName}
                  </div>
                  <div className="col-md-4">
                     {this.course.courseFund}
                  </div>
               </div>
            </div>
         </li>
      )
   }
}
```

```
export default Course
```

b.

```
import React, {Component} from 'react'

class Course extends Component {

  return() {
    render(
      <li className="list-group-item">
        <div >
          <div className="row">
            <div className="col-md-2">
              {props.course.id}
            </div>
            <div className="col-md-6">
              {props.course.courseName}
            </div>
            <div className="col-md-4">
              {props.course.courseFund}
            </div>
          </div>
        </div>
      </li>
    )
  }
}

export default Course
```

c.

```
import React, {Component} from 'react'

class Course extends Component {
   constructor(props) {
      super(props);
   }
   render() {
      return (
         <li key="{this.props.course.id}" className="list-group-item">
            <div >
               <div className="row">
                  <div className="col-md-2">
                     {this.props.course.id}
                  </div>
                  <div className="col-md-6">
                     {this.props.course.courseName}
                  </div>
                  <div className="col-md-4">
                     {this.props.course.courseFund}
                  </div>
               </div>
            </div>
         </li>
      )
   }
}

export default Course
```

d.

```
import React, {Component} from 'react'

class Course extends Component {

   render {
      return (
         <li className="list-group-item">
            <div >
               <div className="row">
                  <div className="col-md-2">
                     {this.props.id}
                  </div>
                  <div className="col-md-6">
                     {this.props.courseName}
                  </div>
                  <div className="col-md-4">
                     {this.props.courseFund}
                  </div>
               </div>
            </div>
         </li>
      )
   }
}

export default Course
```

оја од понудените компоненти користи валидна синтакса за креирање на едноставна компонента WelcomeComponent за приказ на следниот HTML код:

```
<div>
   <b>Welcome</b>
</div>
```

Select one or more:

a.

```
import React, {Class} from 'react'


class WelcomeComponent extends Class {
   return() {
      const message = <h3>Welcome</h3>
      render(
         <span>
            {message}
         </span>
      )
   }
}
export default WelcomeComponent;
```

b.

```
import React from 'react'


class WelcomeComponent  extends React.Component {
   render() {
      const message = <span>Welcome</span>
      return (
         <div>
            {message}
```

```
            </div>
        )
    }
}
export default WelcomeComponent;
```

c.

```
import React from 'react';

const welcomeComponent = (props) => {
    let msg = <b>Welcome</b>
    return (
        <div>
            {msg}
        </div>
    )
}
export default welcomeComponent;
```

d.

```
import React from 'react';

const welcomeComponent = () => {
    let msg = <b>Welcome</b>
    render (
        <div>
            {{msg}}
        </div>
    )
}
export default welcomeComponent;
```

Поврзете ги настаните од животниот циклус на Bean-овите во Spring Container-от.

Повик на методот setBeanFactory на имплементациите на BeanFactoryAware интерфејсот

Answer 1

Поставување на својствата (Populate Properties)

Answer 2

Повик на иницијализациските методи (@PostConstruct)

Answer 3

Извршување на destroy од имплементацииите на DisposableBeans интерфејсот

Answer 4

Извршување на afterPropertiesSet од имплементацииите на InitializingBean интерфејсот Answer 5

Повик на методите за уништување на bean-ovite (@PreDestroy) Answer 6

Иницијализација (Initialize)

Answer 7

Bean-от е креиран и постои во контејнерот

Answer 8

Повик на методот setBeanName на имплементациите на BeanNameAware интерфејсот

Answer 9

Повик на методот setApplicationContext на имплементациите на ApplicationContextAware интерфејсот

Answer 10

Извршување на postProcessAfterInitialization од имплементациите на BeanPostProcessor интерфејсот

Answer 11

Извршување на postProcessBeforeInitialization од имплементациите на BeanPostProcessor интерфејсот

Answer 12

Please answer all parts of the question.

Прв колоквиум

Question **1**
Not yet answered
Marked out of 1.00
Flag question

Доколку серверот успешно враќа JSON како резултат, кои од следните тврдења се точни?

Select one or more:

- a. Одговорот е со `Status Code 404`
- b. Клиентот го детектира типот на податоците од содржината во телото на одговорот
- c. Одговорот е со `Status Code 500`
- d. Одговорот е со `Status Code 200`
- e. Одговорот содржи `ContentType` **параметар** со вредност `application/json`
- f. Одговорот содржи `ContentType` **заглавје** со вредност `application/json`

Give your reasons

-

2
ed
out of
question

Кои од следните настани треба да се извршат за пред да може да се креираат Request и Response објектите кај една апликација во J2EE Container?

Select one or more:

- Креирање на Servlet Context
- Повикување на service() од сервлетите
- Поставување на параметри во Servlet Context
- Читање на web.xml
- Иницијализација на сервлетите

Give your reasons

-

Question **3**
Not yet answered
Marked out of 1.00
Flag question

Кој додава HTML во одговорот?

Select one or more:

- a. Web Server
- b. Container
- c. Servlet

Give your reasons

-

4

ed

out of

question

Доколку сакате да се де-регистрирате за добивање на пораки на одреден `Publisher` на податоци, во кој метод од сервлетот ќе го сторите тоа?

Select one:

- a. `init`
- b. Конструктор
- c. `service`
- d. `doPost`
- e. `doGet`
- f. `destroy`

Give your reasons

-

Question **5**

Not yet answered

Marked out of 1.00

Flag question

Селектирајте дали исказите кои ги опишуваат Attribute во J2EE контејнерот.

Select one or more:

- a. Тип Object
- b. Тип String
- c. Може да се променат во филтрите
- d. Не може да се променат во филтрите
- e. Овозможуваат структурирање на кодот според MVC шаблонот

Give your reasons

-

6

ed

out of

question

Доколку во ServletA повикаме dispatcherToServletB.forward(req, resp), што од следното е точно:

Select one or more:

- a. Во ServletB можеме да ги пристапиме параметрите пратени до ServletA
- b. Контејнерот ќе го повика сервлетот мапиран на servletB.do
- c. Клиентот ќе направи ново барање до /servletB.do
- d. Во ServletB можеме да ги пристапиме Session атрибутите поставени во ServletA
- e. Во ServletB можеме да ги пристапиме Request атрибутите поставени во ServletA

Give your reasons

-

7

Во кои од следните сценарија логично е атрибутите да се чуваат во Request?

Select one or more:

- [ ] a. Јазикот на корисникот
- [ ] b. Контакт информации за страната
- [ ] c. Резултати од пребарување
- [ ] d. Подржани валути за плаќање
- [ ] e. ID на тековната сесија

Give your reasons

-

8

Ист корисник (ист прелистувач) во неколку последователни барања (без затворање на прелистувачот) има пристап ист податок зачуван во:

Select one or more:

- [ ] a. Session
- [ ] b. Cookie
- [ ] c. Request
- [ ] d. ServletContext (application context)
- [ ] e. Query String

Give your reasons

-

9

Доколку сакаме да креираме bean од кој ќе се креира само еднаш и оваа инстанца ќе се користи секаде каде што треба, тогаш треба да го конфигурираме:

Select one:

- ( ) a. scope="prototype"
- ( ) b. scope="context"
- ( ) c. scope="instance"
- ( ) d. scope="singleton"

Give your reasons

-

n 10

ed

l out of

question

Поврзете ги дефинициите со концептите од Aspect Oriented Programming (AOP):

Место во извршувањето на апликацијата каде се вклучува аспектот Choose...

Дефинира која задача и кога ќе се изврши Choose...

Дефинира каде може да се изврши аспектот Choose...

Choose...
Advice
Pointcut
Join Point

Give your reasons

-

n 11

ed

l out of

question

Во кој од подолу предложените слоеви е најлогично да биде следната имплементација:

```
public void employeesCards(List<Employee> employees, HttpServletResponse response)
  throws DocumentException, IOException, SQLException {

 ByteArrayOutputStream baos = new ByteArrayOutputStream();
 IDMultiCardsReport report = new IDMultiCardsReport("id_cards_multi", baos, context, employees);
 report.generate();
  new ResponseWriter(response).writePdf(baos, "id_cards");
}
```

Select one:

- a. Domain Model
- b. Persistence
- c. Presentation
- d. Service
- e. User Interface

Give your reasons

-

12

ed

out of

question

REST контролерот ConsultationSlotApi е мапиран на следната локација: `/api/consultations`. Апликацијата е стартувана на порта 8080. Кои од прикажаните методи успешно ќе го обработат PATCH барањето во следната форма:

```
PATCH http://localhost:8080/api/consultations/{{slotId}}
professorId: petko.petkov
Content-Type: application/x-www-form-urlencoded

roomName=117&dayOfWeek=2&from=18:00:00&to=20:00:00
```

Како опционален параметар во барањето, може да се прати и date којшто треба да се десеријализира во објект од типот LocalDate.

Select one or more:

☐ a.

```
@RequestMapping(value="{pathId}",method = RequestMethod.PATCH)
public ConsultationSlot updateSlot(@PathVariable int slotId,
                                   @RequestHeader String professorId,
                                   @RequestParam String roomName,
                                   @RequestParam(value = "dayOfWeek") Integer dayOfWeek,
                                   @RequestParam(value = "date", required = false) @DateTimeFormat(iso = DateTimeFormat.ISO.DATE) LocalDate date,
                                   @RequestParam("from") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime from,
                                   @RequestParam("to") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime to) {
        return consultationSlotService.updateSlot(slotId, professorId, roomName, DayOfWeek.of(dayOfWeek), date, from, to);
}
```

☐ b.

```
@PatchMapping("/{slotId}")
public ConsultationSlot updateSlot(@PathVariable int slotId,
                                   @RequestHeader String professorId,
                                   @RequestParam String roomName,
                                   @RequestParam(value = "dayOfWeek") Integer dayOfWeek,
                                   @RequestParam(value = "date") @DateTimeFormat(iso = DateTimeFormat.ISO.DATE) LocalDate date,
                                   @RequestParam("from") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime from,
                                   @RequestParam("to") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime to) {
        return consultationSlotService.updateSlot(slotId, professorId, roomName, DayOfWeek.of(dayOfWeek), date, from, to);
}
```

☐ c.

```
@PatchMapping("/{slotId}")
    public ConsultationSlot updateSlot(@PathVariable int slotId,
                                   @RequestHeader String professorId,
                                   @RequestParam String roomName,
                                   @RequestParam(value = "dayOfWeek", required = false) Integer dayOfWeek,
                                   @RequestParam(value = "date", required = false) @DateTimeFormat(iso = DateTimeFormat.ISO.DATE) LocalDate date,
                                   @RequestParam("from") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime from,
                                   @RequestParam("to") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime to) {
        return consultationSlotService.updateSlot(slotId, professorId, roomName, DayOfWeek.of(dayOfWeek), date, from, to);
}
```

☐ d.

```
@PatchMapping("/{slotId}")
public ConsultationSlot updateSlot(@RequestParam int slotId,
                                   @RequestHeader String professorId,
                                   @RequestParam String roomName,
                                   @RequestParam(value = "dayOfWeek", required = true) Integer dayOfWeek,
                                   @RequestParam(value = "date", required = false) @DateTimeFormat(iso = DateTimeFormat.ISO.DATE) LocalDate date,
                                   @RequestParam("from") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime from,
                                   @RequestParam("to") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime to) {
        return consultationSlotService.updateSlot(slotId, professorId, roomName, DayOfWeek.of(dayOfWeek), date, from, to);
}
```

Give your reasons

-

13

d

out of

question

Доколку имаме потреба кон клиентот да вратиме Header-от **"user"** во рамките на мапиран метод во Spring MVC, во кој од следните методи е можно да го поставиме?

Select one or more:

a.

```
@Controller
@RequestMapping("/exam")
public class ExamController {

 @RequestMapping("question")
 public String question(HttpSession session, ServletContext context) {
  ...
```

b.

```
@Controller
@RequestMapping("/exam")
public class ExamController {

 @RequestMapping("question")
 public String question(@Header String session) {
  ...
```

c.

```
@Controller
@RequestMapping("/exam")
public class ExamController {

 @RequestMapping("question")
 public String question(HttpServletRequest request) {
  ...
```

d.

```
@Controller
@RequestMapping("/exam")
public class ExamController {

  @RequestMapping("question")
  public String question(@RequestHeader(name="user") String user) {
    ...
```

e.

```
@Controller
@RequestMapping("/exam")
public class ExamController {

 @RequestMapping("question")
 public String question(HttpServletResponse context) {
  ...
```

Give your reasons

-

14

ed

out of

question

Доколку сакаме барањето кон нашиот сервис да биде

GET http://localhost:8080/exam/answer/12?answer=xyz

и истото треба да го имплементираме во контролерот:

```
@Controller("exam")
@RequestMapping("/exam")
public class ExamController {


}
```

Кое од следните е валидна дефиниција на метод кој ќе се справи со ова барање?

Select one:

a.

```
@RequestMapping(value = "/exam/answer/{id}?answer={answer}", method = RequestMethod.GET)
public String question(@RequestParam Long id, @RequestParam String answer) {
...
```

b.

```
@RequestMapping(value = "/answer/{id}?answer={answer}", method = RequestMethod.GET, produces = "application/json")
public String question(@RequestParam Long id, @RequestParam String answer) {
...
```

c.

```
@RequestMapping(value = "/answer/{id}?{answer}", method = RequestMethod.GET, produces = "application/json")
public String question(@PathVariable Long id, @PathVariable String answer) {
...
```

d.

```
@RequestMapping(value = "/exam/answer/{id}", method = RequestMethod.GET, produces = "application/json")
public String question(@PathVariable Long id, @RequestParam String answer) {
...
```

e.

```
@RequestMapping(value = "/answer/{id}", method = RequestMethod.GET, produces = "application/json")
public String question(@PathVariable Long id, @RequestParam String answer) {
...
```

Give your reasons

-

n 15

ed

l out of

question

Нека во веб апликација е поставена следната конфигурацијана view-resolver:

```
<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
   <property name="prefix" value="/WEB-INF/views/" />
   <property name="suffix" value=".html" />
</bean>
```

Која ќе биде датотеката која ќе се избере од ViewResolver-от за приказ (view template) при повик на question() методот дефиниран во продолжение?

```
@Controller
@RequestMapping("/exam")
public class ExamController {

 @RequestMapping(value = "/question", method = RequestMethod.POST)
 public String question() {
  return "view";
 }
}
```

Select one:

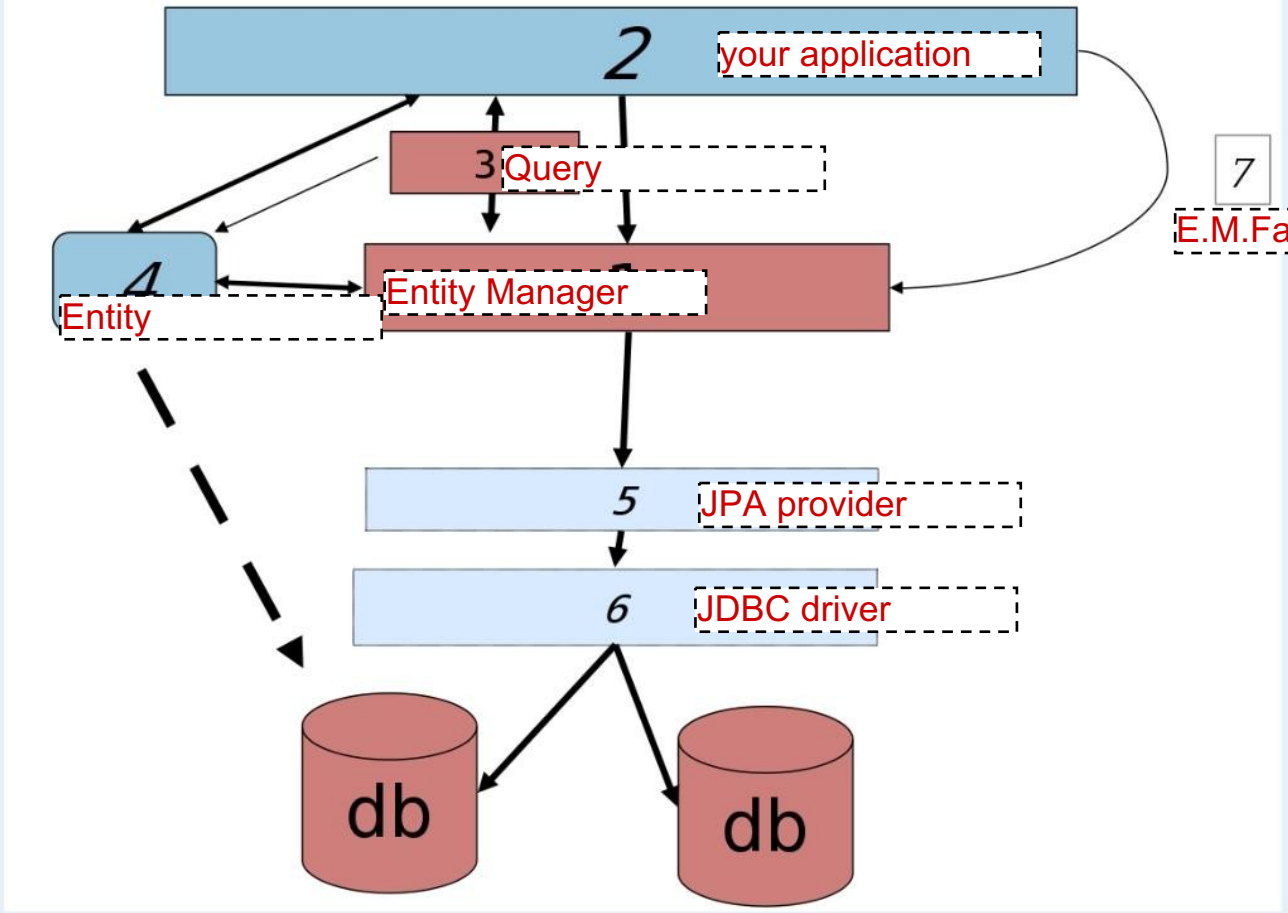
- a. src/main/resources/views/question.html
- b. Нема да се селектира ниту една датотека, ќе се изврши редирекција
- c. src/main/resources/views/view.html
- d. src/main/resources/WEB-INF/views/view.html
- e. src/main/resources/WEB-INF/views/question.html
- f. Нема да се селектира ниту една датотека, ќе се генерира JSON

Give your reasons

-

Втор колоквиум

Поврзете ги компонентите кои одговараат на броевите од сликата.



1 Choose...

2 Choose...

3 Choose...

4 JDBC driver

5 Entity Manager

6 JPA provider

7 Entity

EntityManagerFactory

Query

your applicaton

Choose...

Give your reasons

-

Означете ги JPA анотациите кои треба да се постават за класите да се мапираат во релациона база на податоци.

```
public class Student {
    [2]
    [3]
    public String index;
    [4]
    public String name;
    [5]
    public String email;
    [6]
    public StudyProgram program;
}
```

```
[7]
public class StudyProgram {
    [8]
    public Long id;
    [9]
    public String name;
    [10]
    public List<Student> enrolledStudents;
}
```

[1]

```
@OneToMany(mappedBy="enrolledStudents") [6]
@OneToMany(mappedBy="StudyProgram")
Не мора да има анотација [2, 4, 5, 9]
@ManyToOne(mappedBy="Student")
@Entity [1, 7]
@OneToMany(mappedBy="student")
@OneToMany(mappedBy="program")
@ManyToOne(mappedBy="student")
@Id [3, 8]
@Column(name="x")
@ManyToOne [10]
```

-

Во еден проект е конфигурирано да се користи стратегијата за именување на колоните од аудиторските вежби:

```
spring.jpa.hibernate.naming.implicit-strategy=org.hibernate.boot.model.naming.ImplicitNamingStrategyJpaCompliantImpl
```

Потребно е да ја анотирате класа подолу **со минимален број на JPA анотации** за да се поврзе со табелата **COOPERATE_USER** со следните колони: ("**EMBG**", "**name**", "**job**", "**address**", "**birthDate**").

```
public class CooperateUser {
    [ ]
    public String embg;
    [ ]
    public String name;
    [ ]
    public String job;
    [ ]
    public String address;
    [ ]
    public LocalDate birthDate;
}
```

[ ]

```
@Table(name = "COOPERATE_USER") [1]
@Id @Column(name = "EMBG") [2]
Нема потреба од анотација [3-6]
@Entity
@Id
@Column(name = "EMBG")
@Column(name = "name")
@Column(name = "job")
@Column(name = "address")
@Column(name = "birthDate")
@Column(name = "birth_date")
```

Give your reasons

-

on 4
et
ired
d out of
g question

Доколку го имаме следното маприрање на ентитетите, кои табели ќе се изгенерираат од страна на JPA.

```
@Entity
@Table(name = "STUDENTS")
public class Student extends BaseEntity {

 @Column(name = "student_index")
 public String index;
 public String firstName;
 public String lastName;
 @ManyToMany
 public List<Course> courses;
}

@Entity
@Table(name = "COURSES")
public class Course extends BaseEntity {
 public String name;
 @ManyToMany(mappedBy = "courses")
 public List<Student> students;

}
```

Притоа ја имаме следната конфигурација:

```
spring.jpa.hibernate.naming.implicit-strategy=org.hibernate.boot.model.naming.ImplicitNamingStrategyJpaCompliantImpl
```

Select one or more:

- [ ] a. STUDENT_COURSES
- [ ] b. COURSES_STUDENTS
- [ ] c. STUDENTS
- [ ] d. COURSES

Give your reasons

-

n 5

ed

l out of

question

Дадени се ентитетите:

```
@Entity
public class BookDetails {

    @Id @GeneratedValue
    public Long id;

    public String description;

    @OneToOne
    public Book book;
}

@Entity
public class Book {

    @Id @GeneratedValue
    public Long id;

    public String name;

    public String isbn;

    public Double price;
}
```

Доколку сакаме да ги пронајдеме сите вредности според **isbn** на книгата, кој метод треба да се додаде во следната класа подолу за Spring Data да ни овозможи добивање на валидните резултати:

```
@Repository
public interface BookDetailsRepository extends CrudRepository<BookDetails, Long> {

}
```

Select one:

- a. List<BookDetails> findBylsbn(String isbn);
- b. List<BookDetails> searchIsbn(String isbn);
- c. List<BookDetails> findByBookDetailsIsbnLike(String isbn);
- d. List<BookDetails> findByBookIsbn(String isbn);

Give your reasons

-

6

out of

question

Кој од следниве парчиња код се валидни како REST сервис за добивање на податоци за сите вработени.

Забелешка: Сметајте дека соодветните мапирања и поставувања на ниво на класа и контролер се направени соодветно на потребите.

Select one or more:

a.

```
@GetMapping
private Mono<Employee> getAllEmployees() {
    return employeeRepository.findAllEmployees();
}
```

b.

```
@Bean
RouterFunction<ServerResponse> getAllEmployeesRoute() {
  return route(GET("/employees"),
    req -> ok().body(
      employeeRepository()                    Employee.class));
}
```

c.

```
@GetMapping
private Flux<Employee> getAllEmployees() {
    return employeeRepository
}
```

d.

```
@Bean
RouterFunction<ServerResponse> composedRoutes() {
  return
    route(GET("/employees"),
      req -> ok().body(
        employeeRepository().findAllEmployees(), Employee.class))

    .and(route(GET("/employees/{id}"),
      req -> ok().body(
        employeeRepository().findEmployeeById(req.pathVariable("id")), Employee.class)))

    .and(route(POST("/employees/update"),
      req -> req.body(toMono(Employee.class))
        .doOnNext(employeeRepository()::updateEmployee)
        .then(ok().build())));
}
```

Give your reasons

-

7

out of

question

При веб тестирање, кои се предностите ако се тестира рендерирањето во веб browser-от.

Select one or more:

- [ ] a.
  Well defined API and data structures – resilient
- [ ] b.
  Can test for usability, performance, responsive design, etc.
- [ ] c.
  Unit testing of key modules
- [ ] d.
  Most realistic testing, validates what the end user experiences

Give your reasons

-

8

out of

question

Кој од следните тестови се користи за проверка на точноста на одреден метод, изолиран од неговите зависности?

Select one:

- a. Performance Testing
- b. Integration Testing
- c. Unit Testing
- d. Stress Testing
- e. System Testing
- f. User Acceptance Testing

Give your reasons

-

9

out of

question

Селектирајте ги точните тврдења за процесот на автентикација.

Select one or more:

- [ ] a. Корисникот е валиден доколку е пронајден во User/Credential Storage
- [ ] b. Се идентификува корисникот во рамките на системот
- [ ] c. Се дефинира кои ресурси смее одреден корисник да ги пристапи
- [ ] d. Корисникот секогаш се идентификува според корисничко име и лозинка

Give your reasons

-

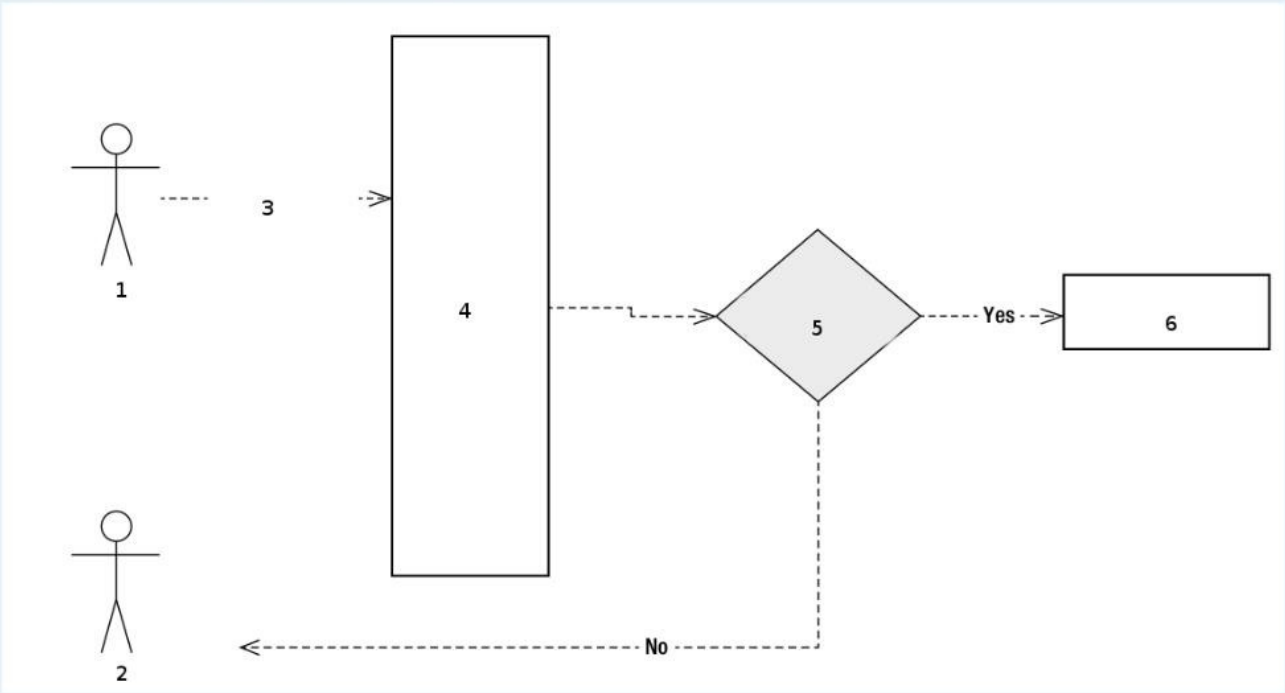
10

d

out of

question

Означете кој термин одговара на бројчето во дијаграмот.



Authorization layer — Choose...

User not allowed access — Choose...

Access Resource — Choose...

Secured Resource — Choose...

Permissions match (user and resource) — Choose...

User — Choose...

4

2

3

6

5

1

Give your reasons

-

ВЕБ ПРОГРАМИРАЊЕ

1. Кој од следниве парчиња код се валидни како REST сервис за добивање на податоци за сите вработени.

Забелешка: Сметајте дека соодветните мапирања и поставувања на ниво на класа и контролер се направени соодветно на потребите.

Select one or more:

a)
```
@GetMapping
private Flux<Employee> getAllEmployees() {
    return employeeRepository.findAllEmployees();
}
```

b)
```
@Bean
RouterFunction<ServerResponse> getAllEmployeesRoute() {
  return route(GET("/employees"),
    req -> ok().body(
      employeeRepository().findAllEmployees(), Employee.class));
}
```

c)
```
@Bean
RouterFunction<ServerResponse> composedRoutes() {
  return
    route(GET("/employees"),
      req -> ok().body(
        employeeRepository().findAllEmployees(), Employee.class))

    .and(route(GET("/employees/{id}"),
      req -> ok().body(
        employeeRepository().findEmployeeById(req.pathVariable("id")), Employee.class)))
```

```
        .and(route(POST("/employees/update"),
          req -> req.body(toMono(Employee.class))
            .doOnNext(employeeRepository()::updateEmployee)
            .then(ok().build())));
    }
```

d)
```
@GetMapping
private Mono<Employee> getAllEmployees() {
    return employeeRepository.findAllEmployees();
}
```

2. Кои од следните тврдења се точни за AuthenticationProvider?

Select one or more:

a. Во иста апликација може да има конфигурирано повеќе од еден AuthenticationProvider
b. AuthenticationProvider служи за најава со Authentication објект
c. AuthenticationProvider го управува процесот на најава кај Spring Security
d. AuthenticationProvider служи за најава само со корисничко име и лозинка

3. Даден е ентитетот:

```
@Entity
@Table(name = "book_details")
public class BookDetails extends BaseEntity {

  @Column(length = 5000)
  public String description;

  @OneToOne
  public Book book;
}
```

Доколку сакаме да ги пронајдеме сите вредности кои имаат одреден опис, кој метод треба да се додаде во следната класа подолу за Spring Data да ни овозможи добивање на валидните резултати:

```
@Repository
public interface BookDetailsRepository extends CrudRepository<BookDetails, Long> {}
```

Select one:

a. **List<BookDetails> findByDescription(String text);**
b. List<BookDetails> searchDetails(String text);
c. List<BookDetails> findByBookDetailsDescriptionLike(String text);
d. List<BookDetails> findByDescriptionLike(String text);

4. Доколку го имаме следното мапирање на ентитетите, кои табели ќе се изгенерираат од страна на JPA.

```
@Entity
@Table(name = "STUDENTS")
public class Student extends BaseEntity {

  @Column(name = "student_index")
  public String index;
  public String firstName;
  public String lastName;
  @ManyToMany
  public List<Course> courses;
}

@Entity
@Table(name = "COURSES")
public class Course extends BaseEntity {

  public String name;
  @ManyToMany
  public List<Student> students;
}
```

Притоа ја имаме следната конфигурација:

```
<property name="jpaVendorAdapter">
    <bean class="org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter"
        p:database="${jpa.database}"
```

```
        p:generateDdl="true"
        p:showSql="false"/>
</property>
```

Select one or more:

a. STUDENTS
b. STUDENTS_COURSES
c. COURSES_STUDENTS
d. COURSES

5. Потребно е да се дефинира нарачка за ресторан. За таа цел е дефиниран ентитетот:

```
public class Order {
  public String orderId;
  public String clientContact;
  public int persons;
  public String table;
  public DateTime date;
}
```

Кој од следните мапирања се валидни за овој ентитет?

Select one or more:

a.
```
@Table(name = "RESTAURANT_ORDER")
public class Order {
@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
public String orderId;

public String clientContact;
public int persons;
public String table;
public DateTime date;
}
```

b.
```
@Table(name = "RESTAURANT_ORDER")
@Entity
public class Order {

  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  public String orderId;

  public String clientContact;
  public int persons;

  @Column(name = "table_number")
  public String table;
  public DateTime date;
}
```

c.
```
@Entity
public class Order {
  public String orderId;
  public String clientContact;
  public int persons;
  public String table;
  public DateTime date;
}
```

d.
```
@Table(name = "RESTAURANT_ORDER")
@Entity
public class Order {

  @Id
  public String orderId;

  public String clientContact;
```

```
public int persons;

@Column(name = "table_number")
public String table;

public DateTime date;
}
```

e.
```
@Entity
public class Order {

 @Id
 public String orderId;

 public String clientContact;
 public int persons;
 public String table;
 public DateTime date;
}
```

6. Кое од следните мапирања на класите Training и Group е валидно?

Select one or more:

a.
```
@Table(name = "TRAINING")
@Entity
public class Training {
 @Id
 @GeneratedValue(strategy = GenerationType.IDENTITY)
 public Long id;
 public String name;
 @OneToMany
 public Group group;
```

```
}

@Table(name = "TRAINING_GROUP")
@Entity
public class Group {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  public Long id;
  public String name;
  @ManyToOne(mappedBy = "group")
  List<Training> trainings;
}
```

b.
```
@Entity
public class Training {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  public Long id;
  public String name;
  @ManyToOne
  public Group group;
}

@Entity
public class Group {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  public Long id;
  public String name;
  @OneToMany(mappedBy = "TRAINING")
  List<Training> trainings;
}
```

c.
```
@Table(name = "TRAINING")
@Entity
public class Training {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  public Long id;
  public String name;
  @ManyToOne
  public Group group;
}

@Table(name = "TRAINING_GROUP")
@Entity
public class Group {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  public Long id;
  public String name;
  @OneToMany(mappedBy = "training")
  List<Training> trainings;
}
```

d.
```
@Table(name = "TRAINING")
@Entity
public class Training {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  public Long id;
  public String name;
  @ManyToOne
  public Group group;
}
```

```
@Table(name = "TRAINING_GROUP")
@Entity
public class Group {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  public Long id;
  public String name;
  @OneToMany(mappedBy = "group")
  List<Training> trainings;
}
```

e.
```
@Table(name = "TRAINING")
@Entity
public class Training {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  public Long id;
  public String name;
  @ManyToOne
  public Group group;
}

@Table(name = "TRAINING_GROUP")
@Entity
public class Group {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  public Long id;
  public String name;
  @Transient
  List<Training> trainings;
}
```

7. Нека во веб апликација е поставена следната конфигурацијана view-resolver:

```
<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="prefix" value="/WEB-INF/views/"/>
    <property name="suffix" value=".jsp"/>
</bean>
```

Која ќе биде датотеката која ќе се избере од ViewResolver-от за приказ (view template) при повик на question() методот дефиниран во продолжение?

```
@Controller
@RequestMapping("/exam")
public class ExamController {
  @RequestMapping(value = "/question", method = RequestMethod.POST)
  public String question() {
    return "question";
  }
}
```

Select one:

a. /views/question.jsp

b. Нема да се селектира ниту една датотека, ќе се генерира JSON

c. Нема да се селектира ниту една датотека, ќе се изврши редирекција

d. /WEB-INF/jsp/question.jsp

e. /WEB-INF/views/question.jsp

8. Во кој од подолу предложените слоеви е најлогично да биде следната имплементација:

```
public List<User> getAllUsers() throws IOException {
  File usersSource = new File(SOURCE_PATH);
  BufferedReader br = new BufferedReader(new InputStreamReader(new FileInputStream(usersSource)));
  CSVFormat format = CSVFormat.DEFAULT.withHeader().withDelimiter(',');
  CSVParser parser = new CSVParser(br, format);
  List<User> users = new ArrayList<User>();
  for (CSVRecord record : parser)
  users.add(parseUser(record));
  parser.close();
  br.close();
  return users;
}
```

Select one:

a. User Interface
b. Service
c. Persistence
d. Domain Model
e. Presentation

9. Кои од следните анотации ги вклучуваат класите во Spring контекстот?Select one or more:

a. @Bean

b. @Service

c. @Singleton

d. @Context

e. @Prototype

f. @Configuration

10. Во кои од следните сценарија логично е атрибутите да се чуваат во Servlet Context(application scope)?

Select one or more:

a. Подржани валути за плаќање

b. ID на тековната сесија

c. Корисничкото име при најава

d. Јазикот на корисникот

e. Резултати од пребарување

f. Контакт информации за страната

11. Доколку сакате да се регистрирате за добивање на пораки на одреден Publisher на податоци, во кој метод од сервлетот ќе го сторите тоа?

Select one:

a. Конструктор

b. doPost

c. service

d. init

e. doGet

f. destroy

12. Различни корисници од различни прелистувачи (browsers) имаат пристап до ист податок зачуван во:

Select one or more:

a. Request

b. Session

c. Query String

d. ServletContext (application context)

e. Cookie

13. Доколку во ServletA повикаме dispatcherToServletB.forward(req, resp), што од следното еточно:

Select one or more:

a. Во ServletB можеме да ги пристапиме Request атрибутите поставени во ServletA

b. Во ServletB можеме да ги пристапиме Session атрибутите поставени во ServletA

c. Контејнерот ќе го повика сервлетот мапиран на servletB.do

d. Во ServletB можеме да ги пристапиме параметрите пратени до ServletA

e. Клиентот ќе направи ново барање до /servletB.do

14. Селектирајте дали исказите опишуваат Attribute или Parameter во J2EE контејнерот.

Тип String **Parameter**

Може да се променат во сервлетите **Attribute**

Не може да се променат во сервлетите **Parameter**

Тип Object **Attribute**

Овозможуваат структурирање на кодот според MVC шаблонот **Attribute**

15. Што од следното е валидно за состојба (state) во React.JS?

Select one or more:

a. Менувањето на вредностите може да се направи директно со доделување на вредност на објектот this.state надвор од конструкторот.

b. Менувањето на вредностите може да се направи директно со користење на функцијата this.setState() надвор од конструкторот.

c. Вредностите кои веќе еднаш се поставени во state на компонента, не може да се менуваат.

d. Вредностите кои веќе еднаш се поставени во state на компонента, може да се менуваат.

e. Во рамките на конструкторот на компонентата не може да се прави доделување навредности на објектот this.state

f. Во рамките на конструкторот на компонентата може да се прави доделување на вредности на објектот this.state

16. Кој од понудените опции прикажува валидна синтакса на функцијата decrement за асинхроно намалување на вредноста за 1 на бројачот во променливата counter сместена во состојбата на компонентата при клик на копчето Decrement? Почетната вредност на бројачот е 100.

render() функцијата изгледа вака:

```
render() {
    return (
        <div className="text-center">
            <label className="control-label">Counter value</label>
            <h3>{this.state.counter}</h3>
            <button className="btn btn-primary" onClick={this.decrement}>Decrement</button>
        </div>
    )
}
```

Функционалноста на компонентата е дадена во следните 2 слики:

counter value 100 decrement

По клик на копчето Decrement, потребно е да го даде следниот излез:

counter value 99 decrement

Select one:

a.

```
decrement => {
    this.setState((state)=> ({
        counter:state.counter-1
    }));
}
```

b.

```
decrement = (e) => {
    this.setState((state)=> ({
        counter:state.counter-1
    }));
}
```

c.

```
decrement {
    this.setState((state)=> ({
        counter:state.counter-1
    }));
}
```

d.

```
decrement() {
    this.state = {
        counter:this.state.counter-1
    }
}
```

e.

```
function decrement(e) {
    this.state.counter--;
}
```

17. Потребно е да се имплементира форма за додавање на нов вработен во некој систем за менаџирање на вработени во рамките на една компанија. Податоците за новиот вработен се додаваат преку соодветна форма. За секој вработен треба да знаеме име, презиме и работен стаж. Сите податоци се внесуваат преку текстуални полиња. Внесувањето на нов вработен се прави со клик на копчето Додади. Податоците за вработените се чуваат во соодветна JSON листа во состојбата на компонентата. Додавањето на нов вработен во листата, треба да биде асинхроно.

Кодот на целосната компонента EmployeeComponent изгледа вака:

```
import React, {Component} from 'react'

class EmployeeComponent extends Component {
    constructor(props) {
        super(props);
        this.state = {
            employees: [
                {
                    name: "Petre",
                    surname: "Petrov",
                    workExperience: 34
                },
                {
                    name: "Petar",
                    surname: "Petrov",
                    workExperience: 12
                }
            ]
        }
    }

    handleSubmit = ?

    render() {
```

```
    let employeesList = this.state.employees.map((item, index) => (
        <li>{item.name}</li>
    ));

    return (
        <div className="container-fluid">
            <ul>
                {employeesList}
            </ul>
            <form onSubmit={this.handleSubmit}>
                <div className="wrapper">
                    <div className="row">
                        <div className="col-md-12">
                            <label className="control-label">Name:</label>
                            <input name={"name"} type={"text"}
                                   className="form-control"/>
                        </div>
                        <div className="col-md-12">
                            <label className="control-label">Surname:</label>
                            <input name={"surname"} type={"text"}
                                   className="form-control"/>
                        </div>

                        <div className="col-md-12">
                            <button type={"submit"} className="btn btn-success">Submit</button>
                        </div>
                    </div>
                </div>
            </form>
        </div>
    )
}
```

```
}

export default EmployeeComponent;
```

Кој од следните имплементации нуди синтаксички валиден код за функцијата handleSubmit кој е соодветен во веќе приложениот код?

Select one:

a.

```
handleSubmit(e) {
    e.preventDefault();
    const employee = {
        name: e.target.name.value,
        surname: e.target.surname.value
    }
    this.setState((state) => {
        return {
            "employees":
                [state.employees,employee]
        }
    });
};
```

b.

```
handleSubmit = (e) => {
    e.preventDefault();
    const employee = {
        name: e.target.name.value,
        surname: e.target.surname.value
    }
    this.setState((state) => {
        return {
            "employees":
                [...state.employees,employee]
```

```
        }
    });
};
```

c.

```
handleSubmit = (e) => {
    e.preventDefault();
    let employee = {
        name: e.target.name.value,
        surname: e.target.surname.value
    }
    this.setState({
            "employees":
                [...state.employees,employee]
        }
    );
};
```

d.

```
handleSubmit(e) {
    e.preventDefault();
    let employee = {
        name: e.target.name.value,
        surname: e.target.surname.value
    }
    this.setState((state) => {
        return {
            "employees":
                [...state.employees,employee]
        }
    });
};
```

18. Избери која од понудените опции дава валидна синтакса во ReactJS за прикажување на детали за предмет во една студиска програма. Името на компонентата е Course. Податоците за курсот се предаваат како својство на следниот начин:

```
let obj = {
     id:1,
    courseName:"WP",
    courseFund:"2+2+1"
};
<Course course={obj}/>
```

Select one:

a.

```
import React, {Component} from 'react'

class Course extends Component {

   render() {
      return (
         <li key="{this.props.course.id}" className="list-group-item">
            <div >
               <div className="row">
                  <div className="col-md-2">
                     {this.course.id}
                  </div>
                  <div className="col-md-6">
                     {this.course.courseName}
                  </div>
                  <div className="col-md-4">
                     {this.course.courseFund}
                  </div>
               </div>
```

```
            </div>
        </li>
      )
   }
}

export default Course
```

b.

```
import React, {Component} from 'react'

class Course extends Component {

   return() {
      render(
         <li className="list-group-item">
            <div >
               <div className="row">
                  <div className="col-md-2">
                     {props.course.id}
                  </div>
                  <div className="col-md-6">
                     {props.course.courseName}
                  </div>
                  <div className="col-md-4">
                     {props.course.courseFund}
                  </div>
               </div>
            </div>
         </li>
      )
   }
}
```

export default Course

c.

```
import React, {Component} from 'react'

class Course extends Component {
   constructor(props) {
      super(props);
   }
   render() {
      return (
         <li key="{this.props.course.id}" className="list-group-item">
            <div >
               <div className="row">
                  <div className="col-md-2">
                     {this.props.course.id}
                  </div>
                  <div className="col-md-6">
                     {this.props.course.courseName}
                  </div>
                  <div className="col-md-4">
                     {this.props.course.courseFund}
                  </div>
               </div>
            </div>
         </li>
      )
   }
}

export default Course
```

d.

```
import React, {Component} from 'react'
```

```
class Course extends Component {


    render {
        return (
            <li className="list-group-item">
                <div >
                    <div className="row">
                        <div className="col-md-2">
                            {this.props.id}
                        </div>
                        <div className="col-md-6">
                            {this.props.courseName}
                        </div>
                        <div className="col-md-4">
                            {this.props.courseFund}
                        </div>
                    </div>
                </div>
            </li>
        )
    }
}


export default Course
```

19. Која од понудените компоненти користи валидна синтакса за креирање на едноставнакомпонента WelcomeComponent за приказ на следниот HTML код:

```
<div>
    <b>Welcome</b>
</div>
```

Select one or more:

a.

```
import React, {Class} from 'react'

class WelcomeComponent extends Class {
    return() {
        const message = <h3>Welcome</h3>
        render(
            <span>
                {message}
            </span>
        )
    }
}
export default WelcomeComponent;
```

b.

```
import React from 'react'

class WelcomeComponent extends React.Component {
    render() {
        const message = <span>Welcome</span>
        return (
            <div>
                {message}
            </div>
        )
```

```
    }
}
export default WelcomeComponent;
```

c.

```
import React from 'react';

const welcomeComponent = (props) => {
    let msg = <b>Welcome</b>
    return (
        <div>
            {msg}
        </div>
    )
}
export default welcomeComponent;
```

d.

```
import React from 'react';

const welcomeComponent = () => {
    let msg = <b>Welcome</b>
    render (
        <div>
            {{msg}}
        </div>
    )
}
export default welcomeComponent;
```

20. Поврзете ги настаните од животниот циклус на Bean-овите во Spring Container-от.

4 Повик на методот setBeanFactory на имплементациите на BeanFactoryAware интерфејсот

2 Поставување на својствата (Populate Properties)

8 Повик на иницијализациските методи (@PostConstruct)

11 Извршување на destroy од имплементациите на DisposableBeans интерфејсот

7 Извршување на afterPropertiesSet од имплементациите на InitializingBean интерфејсот

12 Повик на методите за уништување на bean-овите (@PreDestroy)

1 Иницијализација (Initialize)

10 Bean-от е креиран и постои во контејнерот

3 Повик на методот setBeanName на имплементациите на BeanNameAware интерфејсот

5 Повик на методот setApplicationContext на имплементациите на ApplicationContextAwareинтерфејсот

9 Извршување на postProcessAfterInitialization од имплементациите на BeanPostProcessorинтерфејсот

6 Извршување на postProcessBeforeInitialization од имплементациите на BeanPostProcessorинтерфејсот