# 基本操作

## 問1

下記のTestクラスをコントローラとして利用し、リクエストを送るとindex()メソッドが実行されるようにしたい。しかし、実際にリクエストを送っても、正常に機能しなかった。原因として正しいものはどれか。

[Testjava]

package jp.co.sss.test.controller;

import org.springframework.stereotype.Controller;

@Controller

public class Test {

public String index() {

return "top";

}

}

1. クラス名の末尾に「Controller」と記述されていない。
2. index()メソッドの戻り値に拡張子「.html」が記述されていない。
3. index()メソッドにリクエスト時に実行させるためのアノテーションがない。
4. index()メソッドの戻り値の内容とメソッド名が一致していない。

正解：3

解説

『東京ITスクール Spring』　第2章 基本操作

コントローラとして利用するクラスの名前に「Controller」と記述しなければならないというルールはありません。そのため、選択肢1は不正解です。

ビューに遷移する場合、拡張子の記述は不要です。そのため、選択肢2は不正解です。

メソッド名と戻り値の内容は一致している必要はありません。そのため、選択肢4は不正解です。

リクエストにより実行されるメソッドには@RequestMappingアノテーションを付与する必要があります。したがって、選択肢3が正解です。

## 問2

Spring Bootのプロジェクト内の設定ファイルapplication.propertiesに下記の設定が登録されている。このとき、このプロジェクト内のコントローラにアクセスするためのURLとして正しいものはどれか。

ただし、アクセス対象のコントローラのメソッドは末尾が「/test」のURLでアクセスできるものとする。

[application.properties]

server.servlet.context-path=/school

server.port=15000

1. http://[[ホスト名](about:blank)]/school/test
2. http://[[ホスト名](about:blank)]:15000/test
3. http://[[ホスト名](about:blank)]:15000/school
4. http://[[ホスト名](about:blank)]:15000/school/test

正解：4

解説

『東京ITスクール Spring』　第2章　基本操作

application.properties中の「server.servlet.context-path」でコンテキストパスが指定されます。また、「server.port」でポート番号が指定されます。問題の設定ではコンテキストパスが「/school」、ポート番号が「15000」となっているため、URLは「http://[[ホスト名](about:blank)]:15000/school/〇〇」と記述する必要があります。したがって、選択肢4が正解です。

## 問3

下記のHTMLファイルをSpring Boot環境でのビューとして、かつThymeleafの機能を持たせたい。その場合に必要な操作として誤っているものはどれか。

なお、対象のWebアプリケーションはEclipseのプロジェクト内で実装するものとする。

[top.html]

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Spring 総合試験</title>

</head>

<body>

<h2>トップページ</h2>

</body>

</html>

1. プロジェクトに対してThymeleafを利用可能にするための設定を追加する。
2. 対象のHTMLファイルの拡張子を「xmlns」に変更する。
3. 対象のHTMLファイル中に「<xmlns:th="http://www.thymeleaf.org">というタグを追記する。
4. 対象のHTMLファイルをプロジェクト中の「src/main/resources/templates」以下に保存する。

正解：2

解説

『東京ITスクール Spring』　第2章 基本操作

Thymeleafを利用するにはまず、プロジェクト中のpom.xmlにThymeleafを利用可能にするための設定を追加する必要があります。Eclipseでは専用UIで使用したい機能を追加できるようになっています。そのため、選択肢1は不正解です。

HTMLファイルをThymeleafとして機能させるには、<html>タグに「xmlns:th="[http://www.thymeleaf.org"」という属性を追記する必要があります](about:blank)。そのため、選択肢3は不正解です。

STSでは、ビューは「src/main/resources/templates」以下のフォルダ以下に保存する仕様となっています。そのため、選択肢4は不正解です。

Thymeleafの機能を持たせるにあたって、HTMLファイルの拡張子を変更する必要はありません。したがって、選択肢2が正解です。

# フォーム処理

## 問4

HTMLファイル中の下記フォームを使用して入力値をサーバに送信し、コントローラTestControllerのinput()メソッドを実行したい。

フォームの送信ボタンを押したとき、リクエストURLとして正しいものはどれか。

なお、リクエスト対象のWebアプリケーションにはコンテキストパスの設定が登録されていないものとする。

また、リクエストURL中のポート番号はこの問題では省略している。

[test.html]

・・・

<form th:action="@{/test}" method="GET">

　　　　<input type="text" name="user"/>

　　　　<input type="submit" value="送信" />

</form>

・・・

1. [http://[ホスト名](about:blank)]/TestController/test?user=[入力値]
2. [http://[ホスト名](about:blank)]/index/test?user=[入力値]
3. [http://[ホスト名](about:blank)]/test?user=[入力値]
4. [http://[ホスト名](about:blank)]/test/user/[入力値]

正解：3

解説

『東京ITスクール Spring』　第4章 フォーム処理

<form>タグのmethod属性には「GET」が指定されているため、GET送信でパラメータが送信されます。GET送信では、URLの末尾に「?パラメータ名=入力値」という形式でクエリ情報が付随されます。そのため、選択肢4は形式が異なるため、不正解です。

「[http://[ホスト名](about:blank)]/」の後ろには、コントローラのメソッドを呼び出すために指定された文字列が追記されるかたちとなります。この文字列は<form>タグの「th:action」属性で指定します。この属性により「コンテキストパスを含むパス名」が組み立てられますが、問題の前提として今回はコンテキストパスを設定していないため、問題のフォームからは下記のリクエストURLが送信されることになります。

[http://[ホスト名](about:blank)]/test?user=[入力値]

したがって、選択肢3が正解です。

なお、コントローラのメソッドにリクエストするためのURL中にコントローラやメソッドの名前を記述するルールはありません。そのため、選択肢1と2は不正解です。

## 問5

HTMLファイル中の下記フォームを使用して入力値をサーバに送信し、コントローラTestControllerのinput()メソッドを実行したい。

このとき、input()メソッドを定義する際の作業として、Spring Bootの仕様上誤っているものはどれか。

[test.html]

・・・

<form th:action="@{/test}" method="POST">

　　　　<input type="text" name="user"/>

　　　　<input type="submit" value="送信" />

</form>

・・・

1. input()メソッドにパラメータ名と同名の引数をString型で定義する。
2. input()メソッドにパラメータ名と同名の引数をint型で定義する。
3. リクエスト取得時に実行するためのアノテーションを記述し、「method = RequestMethod.POST」をそのアノテーションの引数として追記する。
4. リクエスト取得時に実行するためのアノテーションを記述し、「method = "POST"」をそのアノテーションの引数として追記する。

正解：4

解説

『東京ITスクール Spring』　第4章　フォーム処理

パラメータを取得する場合、対象のメソッドにはパラメータ名と同名の引数を指定します。このとき、引数の型はString型、int型のどちらでも定義できます。そのため、選択肢1と2はSpringの仕様内容として正しいですが、「誤っているものはどれか」なので不正解となります。

なお、引数の型がintやIntegerなどの数値を扱う型の場合、数値に変換できない文字列がパラメータとして渡されると、エラーが発生します。

問題のフォームはパラメータをPOST送信で送ります。そのため、リクエストを受け取るメソッドにはPOST送信用の設定が必要です。具体的には@RequestMappingアノテーションの引数として下記内容を指定します。

method = RequestMethod.POST

そのため、選択肢3も不正解です。

HTMLでの送信メソッドの指定方法のように「method = “POST”」とは記述できません。したがって、選択肢4が正解です。

## 問6

HTMLファイル中の下記フォームから送信された各パラメータを、Formクラスで受け取れるように実装したい。

今回のケースにおけるFormクラスを定義する際のルールとして、正しいものはどれか。

[test.html]

・・・

<form th:action="@{/test}" method="POST">

<input type="text" name="id" /><br />

<input type="password" name="password" /><br />

<input type="submit" value="送信" />

</form>

・・・

1. Setterメソッドの名前は、必ずsetId、setPasswordと命名する。
2. Formクラスの名前は必ずTestFormと命名する。
3. クラスブロックの手前に必ず@Formアノテーションを付与する。
4. Getterメソッド内には必ずパラメータの値をフィールドに代入する処理を記述する。

正解：1

解説

『東京ITスクール Spring』　第4章　フォーム処理

Formクラスを他の情報と連携した名前で命名しなければならないというルールはありません。そのため、選択肢2は不正解です。

クラスをFormクラスとして機能させる際に、アノテーションの付与は必要ありません。そのため、選択肢3は不正解です。

パラメータの値をフィールドに代入する処理は、Setterメソッドに定義します。Getterメソッドにはフィールドの値を戻りとして返す処理を記述します。そのため、選択肢4は不正解です。

FormクラスのSettterメソッド、Getterメソッドは必ず、set + [先頭を大文字に変換したパラメータ名]、get + [先頭を大文字に変換したパラメータ名]と命名してください。このルールに反すると、パラメータの値の受け渡しが正常にできません。したがって、選択肢1が正解です。

# スコープ

## 問7

コントローラのメソッド内でセッションスコープ中のデータを利用したい。このメソッドの定義として正しいものはどれか

・・・

public String メソッド名() {

HttpSession session = new HttpSession();

}

・・・

・・・

public String メソッド名(HttpSession session) {

・・・

}

・・・

・・・

public String メソッド名(String 引数) {

HttpSession session = 引数;

}

・・・

・・・

public String メソッド名(Session session) {

・・・

}

・・・

正解：2

解説：

『東京ITスクール Spring』　第5章 スコープ

メソッド内でセッションスコープのデータを利用するには、メソッドにHttpSession型の引数を指定することで利用できます。したがって、選択肢2が正解です。

また、@Autowiredアノテーションを付与したHttpSession型のフィールドを定義する方法でも、DIを活用してセッションスコープのデータを利用できます。

その他の選択肢の書き方ではセッションスコープのデータは利用できません。

## 問8

Integer型の変数numberの値をセッションスコープに登録したい。

この処理の記述として正しいものはどれか。

なお、値を保存するスコープ変数名はsessionNumber、セッションスコープの情報を参照している変数はsessionとする。

1. session.getAttribute("sessionNumber", number);
2. session.getAttribute(number, "sessionNumber");
3. session.setAttribute(number, "sessionNumber");
4. session.setAttribute("sessionNumber", number);

正解：4

解説

『東京ITスクール Spring』　第5章 スコープ

セッションスコープに値を保存するにはHttpSession#setAttribute()メソッドを使用します。そのため、選択肢1と2は不正解です。

このメソッドの第1引数にはセッションスコープに登録したいスコープ変数名、第2引数には保存したい値を指定します。

したがって、選択肢4が正解です。

## 問9

Thymeleafの機能が使えるHTMLファイル内で、セッションスコープ中に保存された値を取得して、<p>タグの間にその値を出力したい。編集内容として正しいものはどれか。

なお、スコープ変数の名前はtestとする。

[test.html]

・・・

<body>

<p></p>

</body>

・・・

1. <p var="${session.test}"></p>
2. <p th:text="${session.test}"></p>
3. <p >${session.test}</p>
4. <p >${test}</p>

正解：2

解説

『東京ITスクール Spring』　第5章 スコープ

スコープ中に保存された値をビュー内で取得するには、変数式を使用します。変数式は「${・・・}」という形式で記述します。変数式内には、コープ中のスコープ変数を指定します。セッションスコープ中のスコープ変数testの値を取得する場合は下記のように記述します。

${session.test}

選択肢4は不正解です。

変数式は単体では使用できません。もし、特定のタグの間に値を出力したい場合は、「th:text」という属性の値として変数式を記述します。問題の要件では、<p>タグの間に出力したいため、下記のように編集します。

<p th:text="${session.test}"></p>

したがって、選択肢2が正解です。

その他の選択肢は記述方法が誤っているため、正常に動作しません。

# JPA

## 問10

下記の列を持つemployeeテーブルと関連付けたエンティティEmployeeクラスを定義したい。このとき、emp\_id列と関連付けられたフィールドの定義内容として正しいものはどれか。

なお、id列はシーケンスseq\_empにより、1ずつ加算された連番が自動で登録されるようにしたい。

[employeeテーブル]

列名 型 制約

--------- ----------------- ------------

emp\_id NUMBER(5) PRIMARY KEY

emp\_name VARCHAR2(30 CHAR) NOT NULL

gender NUMBER(1) NOT NULL

@Column

@GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "gen\_emp")

@SequenceGenerator(name = "gen\_emp", sequenceName = "seq\_emp", allocationSize = 1)

private Integer empId;

@Id

@GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "gen\_emp")

@SequenceGenerator(name = "gen\_emp", sequenceName = "seq\_emp", allocationSize = 1)

private Integer empId;

@Id

@GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "gen\_emp")

@SequenceGenerator(name = "emp", sequenceName = "seq\_emp", allocationSize = 1)

private Integer empId;

@Id

@GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "gen\_emp")

@SequenceGenerator(name = "gen\_emp", sequenceName = "sequence\_emp", allocationSize = 1)

private Integer empId;

正解：2

解説

『東京ITスクール Spring』　第6章　データベース操作（CRUD）

主キーであり、シーケンスを利用する列をエンティティのフィールドに紐付ける際の実装方法に関する問題です。

主キーと関連付けられるフィールドには@Idアノテーションを付与します。そのため、選択肢1は不正解です。

シーケンスを利用するには、対象のフィールドに@GeneratedValue（呼び出すジェネレータの種類を指定する）と@SequenceGenerator（使用するシーケンスを指定する）の2種類のアノテーションを付与します。@GeneratedValueのenerator属性と@SequenceGeneratorのname属性の値が一致していることで、対象のシーケンスをりようできるようになります。選択肢3ではこれらの属性の値が一致しないため、シーケンスを呼び出せません。そのため、選択肢3は不正解です。

前述の通り、使用したいシーケンスは@SequenceGeneratoアノテーションで指定します。使用したいシーケンスはsequenceName属性で指定します。今回使用するシーケンスは「seq\_emp」ですが、選択肢4ではsequenceName属性の値が異なります。そのため、選択肢4は不正解です。

選択肢2では、上記の条件を全て満たしています。したがって、選択肢2が正解です。

## 問11

下記の列を持つemployeeテーブルにレコードを登録するためにregistEmployee()メソッドをコントローラに定義した。しかし、下記の内容では期待通りに処理が実行できない。この場合において、不具合の原因ではないものはどれか。

なお、リクエスト送信時にregistEmployee()メソッドが実行される設定は正しく記述されているとする。

また、employeeテーブルを操作するためのリポジトリはEmployeeRepository、employeeテーブルと関連付けられたエンティティはEmployeeとする。

列名 型 制約

--------- ----------------- ------------

emp\_id NUMBER(5) PRIMARY KEY

emp\_name VARCHAR2(30 CHAR) NOT NULL

gender NUMBER(1) NOT NULL

[EmployeeController.java]

・・・

EmployeeRepository repository;

・・・

public String registEmployee() {

　　　　Employee employee = new Employee();

　　　　repository.insert(employee);

　　　　return "employee/regist/regist\_complete";

}

・・・

1. フィールドrepositoryの定義に必要なアノテーションが付与されていない。
2. Employeeのオブジェクトをnew演算子で生成している。
3. エンティティに登録する値が保存されていない。
4. リポジトリが呼び出すメソッド名が誤っている。

正解：2

解説

『東京ITスクール Spring』　第6章 データベース操作（CRUD）

リポジトリを利用する際は、リポジトリに対してDIを機能させるために@Autowiredアノテーションをリポジトリ型のフィールドに付与します。そのため、選択肢1は不正解です。

リポジトリを使用してレコードを登録したい場合は、事前にエンティティのオブジェクトに登録したい値を保存しておく必要があります。そのため、選択肢3は不正解です。

登録処理はsave()メソッドで実行されます。そのため、選択肢4は不正解です。

登録する値をエンティティに保存したい場合、new演算子でオブジェクトを生成して、その中に保存するという方法でも可能です。したがって、選択肢2が正解です。

## 問12

下記の列を持つemployeeテーブルのdept\_idは、departmentテーブルのdept\_idを外部参照している。また、employeeテーブルとdepartmentテーブルは「多対1」の関係にある。

employeeテーブルのエンティティEmployeeを定義するとき、他のエンティティと関連付けるためのフィールドの記述として正しいものはどれか。

なお、departmentテーブルのエンティティの名前は「Department」とする。

[employeeテーブル]

列名 型 制約

--------- ----------------- ------------

emp\_id NUMBER(5) PRIMARY KEY,

emp\_name VARCHAR2(30 CHAR) NOT NULL,

gender NUMBER(1) NOT NULL,

dept\_id NUMBER(2) REFERENCES department(dept\_id)

[departmentテーブル]

列名 型 制約

--------- ----------------- ------------

dept\_id NUMBER(2) PRIMARY KEY

dept\_name VARCHAR2(15 CHAR) NOT NULL

@OneToMany(name = "dept\_id", referencedColumnName = "deptId")

private Department departmentt;

@ManyToOne(name = "dept\_id", referencedColumnName = "deptId")

private Department department;

@OneToMany

@JoinColumn(name = "dept\_id", referencedColumnName = "deptId")

private Department department;

@ManyToOne

@JoinColumn(name = "dept\_id", referencedColumnName = "deptId")

private Department department;

正解：4

解説

『東京ITスクール Spring』　第7章 データベース操作（外部参照）

エンティティ同士の外部参照関係を定義する方法に関する問題です。

外部参照関係にあるテーブルのエンティティにおいて、参照元テーブルのエンティティ内で参照先テーブルのエンティティとの外部参照制約を定義するには@JoinColumnアノテーションを使用します。そのため、選択肢1と2は不正解です。

employeeテーブルとdepartmentテーブルは多対1の関係にあります。

そのため、エンティティEmployee内にエンティティDepartmentと関連付けるためのフィールドを定義する際には、「他対1」をあらわす@ManyToOneアノテーションを付与します。

また、「(name = "dept\_id", referencedColumnName = "deptId")」という記述は@JoinColumnアノテーションの引数として記述します。

したがって、選択肢4が正解です。

# JPQL

## 問13

Oracle Database内にあるemployeeテーブルを操作するために、下記のJPQLを作成した。しかし、下記のJPQLには構文上の誤りがある。誤っている点はどれか。

なお、下記のJPQLではemployeeテーブルに関連付けられたエンティティEmployeeを使用する。

また、エンティティEmployee内には、gender列に関連付けられたフィールドgenderが定義されているとする。

SELECT \* FROM Employee e WHERE e.gender = 1

1. JPQLでの使用が禁止されている句を使用している。
2. FROM句で指定された別名がテーブル名と同名になっていない。
3. WHERE句でパラメータが指定されていない。
4. SELECT句で「\*」を使用している。

正解：4

解説

『東京ITスクール Spring』　第8章 データベース操作（JPQL）

JPQLではSELECT句、FROM句、WHERE句のいずれも利用可能です。そのため、選択肢1は不正解です。

FROM句でエンティティに対して指定する別名は、テーブル名と同名である必要はありません。そのため、選択肢2は不正解です。

WHERE句内ではパラメータを使用せず、固定の値で比較することも可能です。そのため、選択肢3は不正解です。

JPQLのSELECT句では、「全ての列を取得する」という意味を持つ「\*」は使用できません。もし、同様の処理を行いたい場合はエンティティに指定された別名を記述します。

SELECT e FROM Employee e WHERE e.gender = 1

したがって、選択肢4が正解です。

## 問14

JPQLをリポジトリ内に定義する際の正しい記述はどれか。



@NamedQuery()

**public** List<Employee> findEmployee("JPQLの内容");



@NamedQuery("JPQLの内容")

**public** List<Employee> findEmployee();

3.

@Query("JPQLの内容")

**public** List<Employee> findEmployee();

4.

@Query()

**public** List<Employee> findEmployee("JPQLの内容");

正解：3

解説

『東京ITスクール Spring』　第8章 データベース操作（JPQL）

リポジトリ内にJPQLを定義する場合は、@Queryアノテーションを使用します。そのため、選択肢1と2は不正解です。なお、@NamedQueryアノテーションはエンティティ内にJPQLを定義する際に使用します。

また、@Queryアノテーションを使用する場合、JPQLの内容はアノテーションの引数として定義します。

したがって、選択肢3が正解です。

## 

## 問15

Oracle Database内にあるemployeeテーブルの行数を調べたい。そのJPQLをリポジトリ内に定義する場合、正しい記述はどれか。

なお、下記のJPQLではemployeeテーブルに関連付けられたエンティティEmployeeを使用する。

また、エンティティEmployee内には、gender列に関連付けられたフィールドgenderが定義されているとする。



SELECT COUNT FROM Employee employee



SELECT ROWS FROM Employee employee



SELECT COUNT(employee.gender) FROM Employee employee



SELECT ROWS(employee.gender) FROM Employee employee

正解：3

解説

『東京ITスクール Spring』　第8章 データベース操作（JPQL）

行数を取得したい場合は、COUNT関数を使用します。したがって、選択肢2と4は不正解です。

COUNT関数には数える対象となるエンティティまたはフィールドを指定する必要があります。

そのため、選択肢1は不正解です。

したがって、選択肢3が正解です。

# 入力チェック

## 問16

あるFormクラス中のint型のidフィールドに下記2つの入力チェックを行いたい。

idフィールドの定義として最適な記述はどれか。

入力された文字列が空文字ではない（未入力チェック）。

入力された文字列がint型の整数値である（形式チェック）。



@NotBlank

@Integer

private int id;



@NotBlank

@Number

private int id;



@Size(min = 0, max = 10)

private int id;



private int id;

正解：4

解説：

『東京ITスクール Spring』　第9章 入力チェック

@Integerや@Numberといったアノテーションは存在しません。そのため、選択肢1と2は不正解です。

@Sizeアノテーションは文字数の最大値、最小値の制限を設けるためのものです。int型の形式チェックとは用途が異なるため、この場合での利用は不適切です。そのため、選択肢3は不正解です。

フィールドの型がint型の場合、対象のフィールドには形式チェック、および未入力チェックの処理が暗黙的に行われます。そのため、アノテーションの付与は不要となります。

したがって、選択肢4が正解です。

## 問17

入力された数値が上限値を超えた場合にエラーメッセージを出力したい。メッセージリソースに設定する内容として正しいものはどれか。

1. javax.validation.constraints.Max=上限桁数を超えています。

2. javax.validation.constraints.Max.message=上限桁数を超えています。

3. javax.validation.constraints.Max.validationMessage:上限桁数を超えています。

4. javax.validation.constraints.Max.message:上限桁数を超えています。

正解：2

解説

『東京ITスクール Spring』　第10章 メッセージ出力

メッセージリソース内には「キー=値」という形式でメッセージを設定します。そのため、選択肢3と4は不正解です。

また、対象のアノテーションによる入力チェックでのメッセージは「[パッケージ名].[入力チェック用アノテーションを定義するクラス名].message=[メッセージ内容]」という形式で記述します。

そのため、選択肢1は不正解です。

したがって、選択肢2が正解です。

## 問18

下記のエラーメッセージの項目名の箇所（「商品の個数」と記載された箇所）に別のメッセージ（入力項目の名前）を埋め込みたい。また、数値の上限値の箇所（「999」と記載された箇所）にFormクラスで指定された上限値を埋め込みたい。

このとき、メッセージリソースValidationMessages.propertiesに設定するメッセージの内容として最適なものはどれか。

商品の個数が上限値999を超えています。

1. {0}が上限値{value}を超えています。
2. {1}が上限値{value}を超えています。
3. {0}が上限値{max}を超えています。
4. {1}が上限値{max]を超えています。

正解：1

解説

『東京ITスクール Spring』　第10章 メッセージ出力

数値の上限値は@Maxアノテーションでチェックできます。このアノテーションで指定された上限値はメッセージ内で「{value}」と記述して呼び出せます。そのため、選択肢3と4は不正解です。

入力項目の名前は「{0}」と記述するすることで埋め込めます。「{1}」と記述した場合、チェック時の比較対象になる値が埋め込まれるため、今回の要件からは外れます。そのため、選択肢2は不正解です。

したがって、選択肢1が正解です。

# Thymeleaf

## 問19

Thymeleaf機能を持ったHTML内で、スコープ変数usersに格納された複数のユーザ情報を取得して、繰り返し処理で各ユーザ情報を一覧表示したい。

また、合わせて各ユーザ情報と一緒に1から始まる連番も表示したい。

上記の要件を満たす処理として正しいものはどれか。

なお、ユーザ情報のオブジェクトにはフィールドidとnameが定義されており、各ユーザ情報はテーブル形式で出力するものとする。



<tr th:for=*"user, status: ${users}"*>

<td th:text=*"${status.index}"*></td>

<td th:text=*"${user.id}"*></td>

<td th:text=*"${user.name}"*></td>

</tr>



<tr th:for=*"user, status: ${users}"*>

<td th:text=*"${status.count}"*></td>

<td th:text=*"${user.id}"*></td>

<td th:text=*"${user.name}"*></td>

</tr>



<tr th:each=*"user, status: ${users}"*>

<td th:text=*"${status.index}"*></td>

<td th:text=*"${user.id}"*></td>

<td th:text=*"${user.name}"*></td>

</tr>



<tr th:each=*"user, status: ${users}"*>

<td th:text=*"${status.count}"*></td>

<td th:text=*"${user.id}"*></td>

<td th:text=*"${user.name}"*></td>

</tr>

正解：4

解説：

『東京ITスクール Spring』　第12章 Thymeleaf

Thymeleafによる繰り返し処理にはth:each属性を使用します。そのため、選択肢1と2は不正解です。

繰り返し処理の回数をあらわす連番はステータス変数から取得できます。ステータス変数は、繰り返し変数に続けて「,」区切りで定義します。

「ステータス変数.index」と記述すると、0から始まる連番が取得できます。そのため、選択肢3は不正解です。

「ステータス変数.count」と記述すると、1から始まる連番が取得できます。したがって、選択肢4が正解です。

## 問20

Thymeleaf機能を持ったHTML内で、スコープ変数employeeに格納されたオブジェクトが持つidフィールドの値を取得したい。そして、リンクURL式の機能を利用して、idフィールドの値をパラメータとして送信できるようにしたい。

上記の要件を満たす記述として正しいものはどれか。

1. <a th:href="#{/items(id=${employee.id})}" >〇〇</a>

2. <a th:href="@{/items(id=${employee.id})}" >〇〇</a>

3. <a th:href="#{/items/${employee.id}}" >〇〇</a>

4. <a th:href="@{/items/${employee.id}}" >〇〇</a>

正解：2

解説

『東京ITスクール Spring』　第12章 Thymeleaf

リンクURL式は「@{}」と表記します。そのため、選択肢1と3は不正解です。

リンクURL式内でパラメータを指定する場合は、URLの末尾に続けて「(パラメータ名=値)」の形式で記述します。スコープ変数の値をパラメータの値にしたい場合は、変数式を「()」内の値にあたる箇所に記述します。

したがって、選択肢2が正解です。

## 問21

コントローラから個別画面（ビュー）にフォワードすると、個別画面の<body>タグ内の内容をtemplate.htmlに埋め込んだ画面が出力されるようにしたい。

個別画面の内容を埋め込むために、template.htmlに記述する内容として正しいものはどれか。

なお、main.htmlの<body>タグ内の内容は、<div>タグ内に埋め込むものとする。

1. <div id="〇〇" th:replace="${body}"></div>
2. <div id="〇〇" th:replace="~{body :: view}"></div>
3. <div id="〇〇" th:replace="~{layout :: layout(~{::body/content()})}"></div>
4. <div id="〇〇" th:fragment="template(body)"></div>

正解：1

解説

『東京ITスクール Spring』　第12章 Thymeleaf

「th:replace="~{body :: view}」という形式の記述は、ヘッダーやフッターなどの共通部品を埋め込むためのものです。そのため、選択肢2は不正解です。

「th:replace="~{layout :: layout(~{::body/content()})}"」という形式の記述は、ビュー内に記述する内容です。そのため、選択肢3は不正解です。

「th:fragment="template(body)"」という形式の記述は、対象のHTMLファイルをテンプレートとして使用することを明示するためのものです。直接的に個別画面を埋め込むための機能ではないため、選択肢4は不正解です。

個別画面の内容を埋め込むには「th:replace="${body}"」という形式で記述します。「${}」内には個別画面中で埋め込む範囲を指定します。したがって、選択肢1が正解です。