Homework 8

1. Question List
   1. what is IOC

It is a design principle that can be used to reduce the coupling between computer code. The most common method call is DI.

IOC contain two part the one is ApplicationContext and the BeanFacotry. They both are interface. ApplicationContext extend the BeanFactory. ApplicationContext represent the IOC Container.

* 1. what is DI

It is a software design pattern and one of the techniques to be implemented by inversion of control

* 1. what is AOP

The solution for across-cutting concerns. It is a program paradigm which can add additional behavior to existing code, or execute extra methods during the invocation of specific methods.

* 1. what is Aspect, joinpoint, pointcut, advice, target

Aspect: the combination of the pointcut and the advice

Advide: Before, after, around, after throwing, afterreturning

* 1. bean scope

there five type of scope: singleton scope, prototype scope, request, session, and application.

* 1. what is prototype, singleton, request

singleton: globally one been will created for each IOC container

prototype: everytime we do the DI, it will create a new bean

request: each request will create a bean, after request done, it will destroy.

* 1. ApplicationContext vs BeanFactory

IOC contain two part the one is ApplicationContext and the BeanFacotry. They both are interface. ApplicationContext extend the BeanFactory. ApplicationContext represent the IOC Container.

* 1. different types of DI and their pros and cons

setter based: Injected the bean to the current class by using setter method, all annotations to setter method. It is flexible, because it is partially injection.

Constructor based: Using autowired annotation to inject the IOC container, it is safe because it make it easier for unit test. We can inject the mock easily

Field based, all annotations on the field, it is not safe.

* 1. how does @Autowire work

Autowire is work on constructor based injection. It is injected the dependency.

* 1. @Autowire and @Qualifer

Autowire: It is automatically injects a bean into a class field, constructor, or setter.

Qualifer: Specifies which bean should be injected by providing a bean identifier. Apply along with autowired.

Homework 9

1. Question List
   1. Why spring boots? pros and cons

Spring boot has auto-configuration, it can handle all dependencies problem, need starter-validation dependency. It has default dispatch servlet tomcat

* 1. How to start a spring boot project from scratch?

So we can create our project on start.spring.io, then configure the project. After that we can download the zip file. Open our idea to import the unzip file. After that we can create our main class then run it.

* 1. @Controller vs @RestController

@Controller: used for handling web application requests and returning views

@RestController: used for building RESTful web services that return xml, JSON, and other type. It contains controller and requestBody

* 1. @PathVariable vs @RequestParam

@PathVariable: is used to extract values from URI of the request.

@RequestParam: is used to extract query parameters from the request URL

* 1. @RequestBody vs @ResponseBody

@RequestBody: is used to extract the body of an HTTP request sent by the client. Used for post and put

@ResponseBody: get the response from the http request which it java object and transfer into JSON format

* 1. how to use GetMapping, PutMapping, PostMapping, DeleteMapping, RequestMapping?

GetMapping: it is mapping the http get request.

PutMapping: it is mapping the http updated request.

PostMapping: it is mapping the post request.

DeleteMapping: it is used to delete request

RequestMapping: it is used to get specific data

* 1. What is Spring Actuator?

Spring Actuator is a sub-project of SpringBoot that provides production-ready features to help you monitor and manage your application.

* 1. how to achieve async in spring boot application

we can use annotation @EnableAsync, and we also can use the @Async in the method

Homework 10

1. question list
   1. how does spring handle exception

An exception is thrown from the controller, and we need to use DI to injected the ExceptionHandler to the exist same controller, it there is exception it will be handle by @Exception.

* 1. how does spring validate data

Using @NotNull, @NotBlank, @Max, @Min, @Email

* 1. how does spring do the log

By default spring boot log only sent to the console, and does not write the log file. But we can have log dependencies injection to set the log to the console.

* 1. cache hit vs cache miss

A cache hit occurs when requested data is found in the cache, leading quick retrieval, while a cache miss happens when the data is not in the cache and must be retrieved from slower storage

* 1. do some research on Redis and have a basic understanding

Redis is an in-memory store, and it is no-sql database. It used as a database, cache and message broker.

* 1. sql vs no sql database

SQL is relational database which each table has relation, it is best used for structured data. And it has predefine schema. It is vertically scalable.

Nosql database is document type, key-value, graph or wide-columns store. And it has dynamic schema. It is best used for unstructured data. It is horizontally scalable

* 1. what is database normalization

it is limited the redundant data to store efficiently.

* 1. vertical scaling vs horizontal

vertical scaling mean when database store in a machine, if we want to upgrade it we need to upgrade the resources on that machine such as CPU, RAM, Storage.

Horizontal scaling mean we need more server to distribute the workload

* 1. what is ACID

ACID stands for atomicity, consistency, isolation, and durability

Atomicity: Ensures that a transaction is threated as a single, indivisible unit of work

Consistency: Guarantees that a transaction brings the database from one valid state to another valid state.

Isolation: Ensures that concurrent transactions do not interfere with each other

Durability: Once a transaction is committed, the changed are permanently stored and will survive system crashes or failures.

* 1. what is CAP

Consistency

Availability

Partition tolerance

Homework 11

1. question list
   1. jdbc vs hibernate

jdbc is tuomic transaction, write the sql query.

Hibernate: it is object relational mapping

* 1. Statement vs PreparedStatement vs CallableStatement

Statement: used to execute normal sql queries

PreparedStatement: used to execute dynamic or parameterized sql queries

CallableStatement: used to execute the store procedures.

* 1. how to prevent sql injection

used of prepared statement

use of perperly construted storeprocedures.

Akkiw-list input validation

* 1. what is ORM

It is Object relational mapping. a technique to map relational database structure/operations to Java Classes and methods. makes the database operations more object-oriented.

* 1. JPA vs Hibernate

Hibernate is the most popular JAP implementation

Hibernate has all jap method, but has some specific

Session: similar to JAP’s entityManager, created by the session factory, Generates SQL STATEMENTS through java api

Cache: first level session

Second level session

* 1. what are the Persistent State in Entity LifeCycle

The entity object maintains by the session

* 1. Mapping relationship

OneToOne

OneToMany

ManyToOne

ManyToMany

* 1. what is the cascade type

It is when we perform some action on the target entity it will affect the relation entity

There is six cascade type3: Persist, merge, refresh, remove, detach, all

* 1. what is the fetch type

the fetch type is when we fetch data by mapping different relationship. There is two type one is lazy loading and other one is eager loading. So the lazy loading will apply on oneToMany and ManyToMany

And Eager loading will apply on OneToOne, ManyToOne.

* 1. what is the first/second level cache

first level cache is also known as session level cache, which is open by default.

Second level cache is session factory level cache, it is close by default, and it is global.

* 1. left join vs right join vs inner join vs outer join vs cross join

left join: return all records from left table match record from right table

right join: return record from right table match record from left table.

Inner join: return record match in both table

Outer Join: returns all record when there is a match

* 1. union vs union all

union record from both table which condition match, remove the duplicate

union all won’t remove the duplicate