1/31/2020

Kajal Shukla

Experience

It contain interview questions of various company.

Company Name: LeewayHerz

Question Type: Practical Question

1. Implementation of Stack (push and pop) method using array in Java?

Solution:

package ComapniesQuestion;

public class stack {

static final int MAX=1000;

int top;

int a[]=new int[MAX];

boolean isEmpty(){

return(top<0);

}

stack(){

top=-1;

}

boolean push(int x){

if(top>=(MAX-1))

{

System.out.println("Stack overflow");

return false;

}

else{

a[++top]=x;

System.out.println(x+"pushed into stack");

return true;

}}

int pop(){

if (top<0)

{

System.out.println("Stack Underflow");

return 0;

}

else{

int x= a[top--];

return x;

}

}

int peek(){

if(top<0){

System.out.println("Stack Underflow");

return 0;

}

else {

int x=a[top];

return 0;

}

}

}

Company Name: Oodels

Question Type: Practical Question & Theoritical Question

Practical Questions

1. Draw a pascal’s triangle

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

Solution:

package pattern.programs;

public class PascalNumericTriangle {

public static void main(String[] args) {

int i,j,row=5;

for ( i = 0; i < row; i++) {

int number=1;

System.out.printf("%" +(row-i)\*2+ "s", "");

for ( j = 0; j <=i; j++) {

System.out.printf("%4d",number);

number=number\*(i-j)/(j+1);

}

System.out.println();

}

}

}

1. Draw a right pascal’s triangle

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

Solution:

1. Draw a diamond star pattern

\*

\* \*

\* \*

\* \*

\* \*

\* \*

\* \*

\* \*

\*

Solution:

Theoritical Questions:

1)What is advantage of Encasulation?

Ans:

***Encapsulation in Java binds data and code together***. It is defined as the process in which we wrap the data into a single unit. It basically creates a shield and the code cannot be accessed outside the shield or by any code outside the shield. Java Beans class is the example of a fully encapsulated class.

* The [***variables in Java***](https://data-flair.training/blogs/variables-in-java/) or the method of the class are hidden from any other class and cannot be accessed outside the class.
* We can also call it, as Data-hiding.
* The encapsulated class is easy to test.
* Standard IDE’s like Eclipse, Netbeans are providing the facility to generate getter setter methods, so it is very easy to create an encapsulated call
* It can be achieved by declaring the class as private while the methods as public so that the variables can be accessed.

Advantages of Encapsulation in Java

These are benefits of Encapsulation in Java:

* **Data Hiding –** It can provide the programmer to hide the inner classes and the user to give access only to the desired codes. It allows the programmer to not allow the user to know how [variables](https://en.wikipedia.org/wiki/Variable_(computer_science)) and data store.
* **Getter and Setter Methods –**Private member can only be accessed within the same class. An outside class can not access the data members of that class. If you need to access these variables, you have to use public “getter” and “setter” methods.
* **Flexibility –** With this, we can make the data as read-only or write-only as we require it to be. It also improves the maintainability and flexibility of code.
* **Reusability –** It allows the user to a programmer to use the existing code again and again in an effective way.
* **Testing of the code –** Ease of testing becomes easy. So it is better for Unit testing

What is Inheritance?

Ans :

**Inheritance** is a mechanism in which one class acquires the property of another class. For example, a child inherits the traits of his/her parents. With inheritance, we can reuse the fields and methods of the existing class.

What is multi-threading explain with code?

Ans:

 A thread is a light-weight smallest part of a process that can run concurrently with the other parts(other threads) of the same process. Threads are independent because they all have separate path of execution that’s the reason if an exception occurs in one thread, it doesn’t affect the execution of other threads. All threads of a process share the common memory. **The process of executing multiple threads simultaneously is known as multithreading.**

## Creating a thread in Java

There are two ways to create a thread in Java:  
1) By extending Thread class.  
2) By implementing Runnable interface.

* getName(): It is used for Obtaining a thread’s name
* getPriority(): Obtain a thread’s priority
* isAlive(): Determine if a thread is still running
* join(): Wait for a thread to terminate
* run(): Entry point for the thread
* sleep(): suspend a thread for a period of time
* start(): start a thread by calling its run() method

### Method 1: Thread creation by extending Thread class

class MultithreadingDemo extends Thread{

public void run(){

System.out.println("My thread is in running state.");

}

public static void main(String args[]){

MultithreadingDemo obj=new MultithreadingDemo();

obj.start();

}

}

### Method 2: Thread creation by implementing Runnable Interface

class MultithreadingDemo implements Runnable{

public void run(){

System.out.println("My thread is in running state.");

}

public static void main(String args[]){

MultithreadingDemo obj=new MultithreadingDemo();

Thread tobj =new Thread(obj);

tobj.start();

}

}

Difference between abstraction and interface

Ans:

Abstract class and interface both are used to achieve abstraction where we can declare the abstract methods. Abstract class and interface both can't be instantiated.

|  |  |
| --- | --- |
| **Abstract class** | **Interface** |
| 1) Abstract class can **have abstract and non-abstract** methods. | Interface can have **only abstract** methods. Since Java 8, it can have **default and static methods** also. |
| 2) Abstract class **doesn't support multiple inheritance**. | Interface **supports multiple inheritance**. |
| 3) Abstract class **can have final, non-final, static and non-static variables**. | Interface has **only static and final variables**. |
| 4) Abstract class **can provide the implementation of interface**. | Interface **can't provide the implementation of abstract class**. |
| 5) The **abstract keyword** is used to declare abstract class. | The **interface keyword** is used to declare interface. |
| 6) An **abstract class** can extend another Java class and implement multiple Java interfaces. | An **interface** can extend another Java interface only. |
| 7) An **abstract class** can be extended using keyword "extends". | An **interface** can be implemented using keyword "implements". |
| 8) A Java **abstract class** can have class members like private, protected, etc. | Members of a Java interface are public by default. |
| 9)**Example:** public abstract class Shape{ public abstract void draw(); } | **Example:** public interface Drawable{ void draw(); } |

Company Name: Vanisb

Question Type: Theoritical Question

Therotical Question:

What is payment gateway method in e-commerce website?

Ans:

**Payment Gateways** are software and servers that transmit Transaction information to Acquiring Banks and responses from Issuing Banks (such as whether a transaction is approved or declined). Essentially, **Payment Gateways** facilitate communication within banks.

What is add to chart method in e-commerce website?

Ans:

Add to Cart is a way to create a [temporary list](http://ask.mysapl.org/a.php?qid=120301) of items by adding them to your cart, which will keep track of the items until you leave our website.

You can [export](http://ask.mysapl.org/a.php?qid=120303) items in your cart by saving the list to a file or sending it to an email address. You can also place the items on [hold](http://ask.mysapl.org/a.php?qid=120273) or add them to your [wish list](http://ask.mysapl.org/a.php?qid=213363).

Step 1 : Create a table cart and insert record when user clicks on add to cart option.

Step 2: Create a bean named cart.

Step 3: When user clicks on add to cart set data in java bean and fire insert query to add data to cart table.

Step 4: If you want to show items added in cart,fire a query and retrieve data mapped to that user in cart table and show cart count.

Step 5: Once user clicks on cart count.Display all the items in detail by retrieving data from cart table.

Company Name: RawCube

Question Type: Practical Question & Therotical Question

Practical Question:

Write a program to swap Two number without using third variable?

Solution:

Theortical Question:

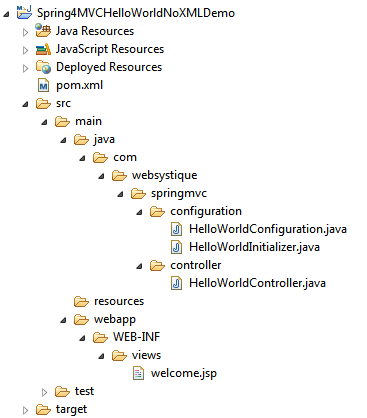
What is spring configration process?

Ans:

#### Step 1: Create a project with required directory structure

Post [Creating a maven web project with eclipse](http://websystique.com/maven/create-a-maven-web-project-with-eclipse/) contains step-by-step instruction to create a maven project with eclipse.

Following will be the final Project structure.



#### Step 2: Update pom.xml with Spring and Servlet dependency

The Spring java-based configuration we are going to discuss depends on Servlet 3.0 api, so we need to include that as a dependency in pom.xml

#### Step 3: Add Controller

Add a controller class under src/main/java

#### Step 4: Add View

Create a new folder named views under WEB-INF and add in a Simple JSP page welcome.jsp (WEB-INF/views/welcome.jsp) to simply access the model value sent from controller.

#### Step 5: Add Configuration Class

Add the below mentioned class under src/main/java with specified package as shown below. This configuration class can be treated as a replacement of spring-servlet.xml as it contains all the information required for component-scanning and view resolver.

#### Step 6: Add Initialization class

Add an initializer class implementing WebApplicationInitializer under src/main/java with specified package as shown below(which in this case acts as replacement of any spring configuration defined in web.xml). During Servlet 3.0 Container startup, this class will be loaded and instantiated and its onStartup method will be called by servlet container.

#### Step 7: Build and Deploy the application

One thing to keep in mind that the Spring java based configuration api’s like WebApplicationInitializer depends on Servlet 3.0 containers.So make sure you don’t have any web.xml with servlet declaration less than 3.0. For our case, we have removed web.xml file from our application.

What is Encapsulation?

Ans:

***Encapsulation in Java binds data and code together***. It is defined as the process in which we wrap the data into a single unit. It basically creates a shield and the code cannot be accessed outside the shield or by any code outside the shield. Java Beans class is the example of a fully encapsulated class.

Diffrence between abstraction and interface?

Ans : Abstract class and interface both are used to achieve abstraction where we can declare the abstract methods. Abstract class and interface both can't be instantiated.

|  |  |
| --- | --- |
| **Abstract class** | **Interface** |
| 1) Abstract class can **have abstract and non-abstract** methods. | Interface can have **only abstract** methods. Since Java 8, it can have **default and static methods** also. |
| 2) Abstract class **doesn't support multiple inheritance**. | Interface **supports multiple inheritance**. |
| 3) Abstract class **can have final, non-final, static and non-static variables**. | Interface has **only static and final variables**. |
| 4) Abstract class **can provide the implementation of interface**. | Interface **can't provide the implementation of abstract class**. |
| 5) The **abstract keyword** is used to declare abstract class. | The **interface keyword** is used to declare interface. |
| 6) An **abstract class** can extend another Java class and implement multiple Java interfaces. | An **interface** can extend another Java interface only. |
| 7) An **abstract class** can be extended using keyword "extends". | An **interface** can be implemented using keyword "implements". |
| 8) A Java **abstract class** can have class members like private, protected, etc. | Members of a Java interface are public by default. |
| 9)**Example:** public abstract class Shape{ public abstract void draw(); } | **Example:** public interface Drawable{ void draw(); } |

Tell me Encapsulation in the room?

Ans:

Any things that exit in the room are Encapsulation because I don’t know about it.

Diffrence between print and println?

Ans:

| **PRINTLN()** | **PRINT()** |
| --- | --- |
| It adds new line after the message gets dispalyed. | It does not add any new line. |
| It can work without arguments. | This method only and only works with argument, otherwise it is an syntax error. |

What is inheritance?

Ans:

**Inheritance** is a mechanism in which one class acquires the property of another class. For example, a child inherits the traits of his/her parents. With inheritance, we can reuse the fields and methods of the existing class.

Institute Name: JLJ Group

Question Type: Practical Question & Theortical Question

Practical Question:

write a program to find largest number in three number.

Solution:

write a program to print table using while loop.

Solution:

write a string program.

Solution:

write a program to find all arm strong number between 0 to 9.

Solution:

Theortical Question:

What is use of scanner class?

Ans:

Scanner is a class in java.util package used for obtaining the input of the primitive types like int, double, etc. and strings. It is the easiest way to read input in a Java program,

What is use of util package?

Ans:

Java.util package contains the collections framework, legacy collection classes, event model, date and time facilities, internationalization, and miscellaneous utility classes. This reference will take you through simple and practical methods available in java.util package.

What is use of nextInt method?

Ans:

Statement **n = s.nextInt();** is used to input the value of an integer variable 'n' from the user.

What is Object Class?

Ans:

The **Object class** is the parent class of all the classes in java by default. In other words, it is the topmost class of java.

The Object class is beneficial if you want to refer any object whose type you don't know. Notice that parent class reference variable can refer the child class object, know as upcasting.

Why we use class in java?

Ans:

Basically classes help you take all the properties and behaviors of an object in your program, and combine them into a single interface, then re-use that interface wherever you need that type of object in your program.