Java Keywords
Key Concepts
Key Concepts #2
OOPS in java
Java Collections#1
Java Collections #2
Exceptions #1
Exceptions #2
Threads#1
Threads#2
InnerClass
More InnerClass
Serialization
Immutable Class
Cloning in Java
Garbage Collection
JSP #1
JSP #2
Programming Q#1
Programming Q#2
Programming Q#3

Java Exception Interview Questions

Q16) Once the control switches to the catch block does it return back to the try block to execute the balance code?

Ans) No. Once the control jumps to the catch block it never returns to the try block but it goes to finally block(if present).

Q17) Where is the clean up code like release of resources is put in try-catch-finally block and why?

Ans) The code is put in a finally block because irrespective of try or catch block execution the control will flow to finally block. Typically finally block contains release of connections, closing of result set etc.

Q18) Is it valid to have a try block without catch or finally?

Ans) NO. This will result in a compilation error. The try block must be followed by a catch or a finally block. It is acceptable to omit the either catch or the finally block but not both.

Q21) How do you get the descriptive information about the Exception occurred during the program execution?

Ans) All the exceptions inherit a method printStackTrace() from the Throwable class. This method prints the stack trace from where the exception occurred. It prints the most recently entered method first and continues down, printing the name of each method as it works its way down the call stack from the top.

Q23)Why is not considered as a good practice to write a single catch all handler to catch all the exceptions?

Ans) You can write a single catch block to handle all the exceptions thrown during the program.

If you use the Superclass Exception in the catch block then you will not get the valuable information about each of the exception thrown during the execution, though you can find out the class of the exception occurred. Also it will reduce the readability of the code as the programmer will not understand what is the exact reason for putting the try-catch block.

Q24) What is exception matching?

Ans) Exception matching is the process by which the the jvm finds out the matching catch block for the exception thrown from the list of catch blocks. When an exception is thrown, Java will try to find by looking at the available catch clauses in the top down manner. If it doesn't find one, it will search for a handler for a supertype of the exception. If it does not find a catch clause that matches a supertype for the exception, then the exception is propagated down the call stack. This process is called exception matching.

Q25) What happens if the handlers for the most specific exceptions is placed above the more general exceptions handler?

Ans) Compilation fails. The catch block for handling the most specific exceptions must always be placed above the catch block written to handle the more general exceptions.

```
//The code below will not compile.
try {
    // code that can throw IOException or its subtypes
} catch (IOException e) {
// handles IOExceptions and its subtypes
} catch (FileNotFoundException ex) {
    // handle FileNotFoundException only
}

// The code below will compile successfully
try {
  // code that can throw IOException or its subtypes
} catch (FileNotFoundException ex)  {
// handles IOExceptions and its subtypes
} catch (IOException e){
// handle FileNotFoundException only
}
```

Q26) Does the order of the catch blocks matter if the Exceptions caught by them are not subtype or supertype of each other?

Ans) No. If the exceptions are siblings in the Exception class's hierarchy i.e. If one Exception class is not a subtype or supertype of the other, then the order in which their handlers(catch clauses) are placed does not matter.

Q27) What happens if a method does not throw an checked Exception directly but calls a method that does? What does 'Ducking' the exception mean?

Ans) If a method does not throw an checked Exception directly but calls a method that throws an exception then the calling method must handle the throw exception or declare the exception in its throws clause. If the calling method does not handle and declares the exception, the exceptions is passed to the next method in the method stack. This is called as ducking the exception down the method stack.

e.g. The code below will not compile as the getCar() method has not declared the CarNotFoundException which is thrown by the getColor () method.

```
void getCar() {
 getColor();
}
void getColor() {
 throw new CarNotFoundException();
}
//Fix for the above code is
void getCar() throws CarNotFoundException {
   getColor();
}
void getColor() {
  throw new CarNotFoundException();
}
```

Q28) Is an empty catch block legal?

Ans) Yes you can leave the catch block without writing any actual code to handle the exception caught

e.g. The code below is legal but not appropriate, as in this case you will nt get any information about the exception thrown.

Q29)Can a catch block throw the exception caught by itself?

Ans) Yes. This is called rethrowing of the exception by catch block.

e.g. the catch block below catches the FileNotFound exception and rethrows it again.

```
void checkEx() throws FileNotFoundException {
  try{
    //code that may throw the FileNotFoundException

  }catch(FileNotFound eFnf){
    throw FileNotFound();
  }
}
```

Recommend Reading

- How java manages Memory?
- Why is it advised to use hashcode and equals together?
- Comparable and Comparator in java
- How to create Singleton class in Java?
- Difference betwen equals and ==?
- When to use abstract class over interface?
- Difference between final, finally and finalize
- Why is it recommended to ues Immutable objects in Java