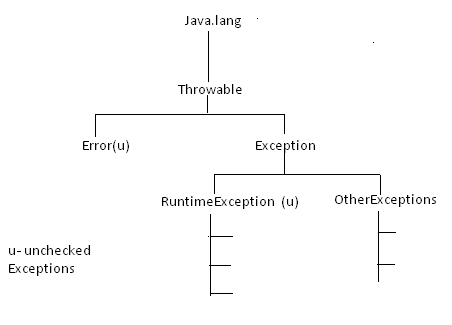
**Exceptions**



1. What is Exception handling?

Ans: An exception is an event which occurs during the execution of a program that disrupts the normal flow of the program’s instructions

2. What is the purpose of Exception handling?

Ans: The main purpose of exception handling is to avoid the abnormal termination of a program.

3. What is the meaning of Exception Handling?

Ans. Exception Handling doesn’t mean repairing an Exception; we have to define alternative way to continue rest of the code normally.

Example: If our programming requirement is to read the data from the file locating at London but at Runtime if London file is not available then we have to use local file alternatively to continue rest of program normally. This is nothing but Exception Handling.

4. What is Throwable?

Ans: Throwable is a class that represents all the errors and exceptions which may occur in java

5. Which is the super class of all exceptions?

Ans: Exception

6. What are checked exceptions and unchecked exceptions?

Ans: The exceptions that are checked at compile-time by the java compiler are called checked exceptions.

Ans: The exceptions that are checked by the JVM are called unchecked exceptions.

7. What is the difference between an exception and an error?

|  |  |
| --- | --- |
| Exception | Error |
| An exception is an error which can be handled, it means when an exception happens the developer can do something to avoid any harm. | An error is an error which cannot be handled, i.e. It happens and the programmer cannot do anything. |
| These are mostly caused by our program and are recoverable. | These are not caused by our program, mostly caused by lake of system resources. These are non recoverable. |

8. Give at least two unchecked and checked exceptions available in java?

Ans: Checked exceptions:

1. SQLException

2. ClassNotFoundException

3. IOException

Etc,,,,

Ans: Unchecked exceptions:

1. ArthemeticException

2. FileNotFoundException

3. NullPointerException

Etc,,,

9. How you handle exceptions in java?

Ans: Exception can be handled by using try and catch block.

10. What is the purpose of try?

Ans: We should maintain all risky code inside the try block.

11. What is the purpose of catch block?

Ans: We have to maintain all Exception Handling code inside the catch block.

12. Is try with multiple catch blocks is possible?

Ans: Yes, A single try block can have any number of catch blocks.

13. If try with multiple catch block present is order of catch blocks important in which order we have to take?

Ans: If try with multiple catch block present then the order of catch block is very important it should be from child to parent but not from parent to child.

14. What are various methods to print Exception information? and differentiate them.

Ans: Throwable class defines the following method to print exception or error information .

**1. printStackTrace() :-** This method print exception information in the following format.

**Name of the Exception: Description StackTrace**

**2. toString():-** This method print exception information in the following format.

**Name of the Exception: Description**

**3. getMessage():-** This method prints only description of the exception.

**Description**

15. If an exception rose inside catch block then what will happen?

Ans: If an exception rose inside catch block and it is not part of any try block then it is always abnormal termination.

16. Is it possible to take try, catch inside try block?

Ans: Yes, It is possible to take try, catch inside try block. That is nesting of try catch is possible.

17. Is it possible to take try, catch inside catch block?

Ans: Yes, It is possible to take try, catch inside catch block.

18. Which package contains exception handling related classes?

Ans: Java.lang

19. What are the two types of Exceptions?

Ans: Checked exceptions and unchecked exceptions.

20. What is User defined Exceptions?

Ans: The exceptions defined by the programmer based on the business logic that are not available in java are called user defined exceptions.

21. Can a catch block exist without a try block?

Ans: No, A catch block should always go with a try block.

22. Can a finally block exist with a try block but without a catch?

Ans: Yes, The following are the combinations try/catch or try/catch/finally or try/finally.

23. What is the importance of finally block in exception handling?

Ans: Finally block will be executed, no matter whether the exception is raised or not, generally finally block is used to close the open resources like dbconnection, file object etc.

24. In which situation finally block will not executed?

Ans: There is one situation where the finally block won’t be executed if we are using system.exit(0) explicitly then JVM itself will be shutdown and there is no chance of executing finally block.

25. If return statement present inside try is finally block will be executed?

Ans: Yes, if return statement present inside try, then also finally block will be executed. Finally block will dominate return statement also.

26. Is it possible to write any statement between try-catch and finally?

Ans: No, it is not possible to write any statement between try catch and finally. If we will try to write any statement between them then we will get compile time error.

27. Is it possible to take two finally blocks for the same try?

Ans: No, it is not possible to take two finally blocks for the same try. If we try to take then we will get compile time error.

28. Is syntax try-finally-catch is valid?

Ans: No, this syntax is not valid. It should be like try-catch-finally then only code will compile.

29. What will happen to the Exception object after exception handling?

Ans: The exception object will be garbage collected.

30. What is the purpose of throw?

Ans: Sometimes we can create Exception object explicitly and we can handover that exception object to the JVM explicitly by throw keyword.

The purpose of throw keyword is to handover our created exception object explicitly to the JVM.

|  |
| --- |
| In this case ArithmeticException object created implicitly and handover to the JVM automatically by the main method. |
| Example1:  class Test{  public static void main(String[] args){  System.out.println(10/0);  }  } |
| In this case creation of an exception object and handover to the JVM explicitly by the programmer. |
| Example2:  Class Test{  Public static void main(String[] args){  Throw new ArithmeticException(“/by Zero”);  }  } |

31. Is it possible to throw an Error?

Ans: Yes, It is possible to throw any Throwable type including Error.

32. Is it possible to throw any java object?

Ans: No, we can use throw keyword only for Throwable objects otherwise we will get compile time error saying incompatible type.

33. After throw is it allow taking any statement directly?

Ans: After throw statement we are not allow to place any statement directly violation leads to compile time error saying Unreachable Statement.

34. What is the purpose of throws?

Ans: The main purpose of throws keyword is to delegate the responsibilities of exception handling to the caller. It requires in the case of checked exception.

35. What is the difference between throw and throws?

Ans: Throw is used when the programmer wants to throw an exception explicitly and wants to handle it using catch block.

Ans: Throws is used when the programmer doesn’t want to handle exception and throw it out of a method.

36. Explain Exception Handling keyword?

Ans: Exception handling keyword:

* **Try :-** To maintain Risky code.
* **Catch:-** To maintain Exception Handling code.
* **Finally:-** To maintain the clean up code.
* **Throw:-** To handover our created exception object to the JVM explicitly.
* **Throws:-** To delegate the responsibilities of Exception Handling to the caller.

37. All the built in exceptions are of what type?

Ans: All the built in exceptions are of type: Checked exceptions.

38. What is the difference between final, finally and finalize?

Ans: Final: The final keyword is used declaring variables as constant.

Ans: Finally: Finally block will be executed, no matter whether the exception is raised or not, generally finally block is used to close the open resources like dbconnection, file object etc.

Ans: Finalize: This method is used for garbage collection. This method will get execute before Garbage Collection is done, because it release the resource which are hold by the program.

39. What is the difference between partially checked and fully checked Exception?

Ans: A checked exception is said to be fully checked if and only if all the child classes also checked otherwise it is called partially checked exception.

Example:

* IOException:- fully checked exception
* Exception:- partially checked exception
* Throwable:- partially checked exception
* RuntimeException:- unchecked exception

40. What is a customized Exception?

Ans: Sometimes based on our programming requirement we have to create our own exception such type of exception is called customized Exception.

Example:

TooYoungException

TooOldException

InsufficientFundException

41. In how many ways we can write a custom exception and explain?

Ans: In two ways

1) Override the getmessage () method by returning the meaningful string message.

2) Create a constructor which accepts the string message and within this constructor pass the message.

42. Explain control flow in try, catch, finally.

Ans:

Try {

Statement1;

Statement2;

Statement3;

}

Catch(X e) {

Statement4;

}

Finally {

Statement5;

}

Statement6;

Case1:

**If there is no Exception then output is**

Statement1

Statement2

Statement3

Statement5

Statement6

**Normal termination**

Case2:

If an exception rose at statement2 and corresponding catch block has matched then output is

Statement1

Statement4

Statement5

Statement5

**Normal termination**

Case3:

An exception rose at statement2 and corresponding catch has not matched then output is

Statement1

Statement5

**Abnormal termination**

Case4:

An exception occurs at statement4 it always abnormal termination but before that finally block will be executed and output is

Statement1

Statement2

Statement5

**Abnormal termination**

Case5:

If an exception rose at statement5 or statement6, it is always abnormal termination.