Experiment No. 6

Title: To implement Exception Handling

Batch:SY-IT(B3) **Roll No.: 16010423076** **Experiment No.: 6**

Aim: Write a program which accepts age of the candidate as a command line argument and checks whether it is within the range (between 18 to 75) or not. If it is within the range then it displays "Candidate satisfy the age criteria", if not then it throws the exception of user defined class "AgeOutOfRangeException".

Resources needed: Java

Theory:

An exception is a problem that arises during the execution of a program. An exception can occur for many different reasons, including the following:

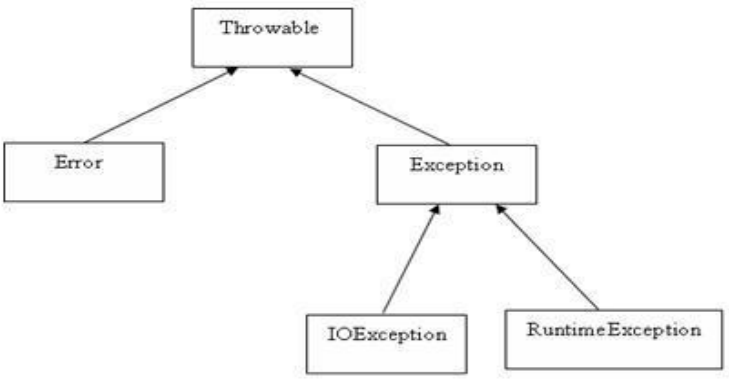
- A user has entered invalid data.
- A file that needs to be opened cannot be found.
- A network connection has been lost in the middle of communications or the JVM has run out of memory.

To understand how exception handling works in Java, you need to understand the three categories of exceptions:

- Checked exceptions: A checked exception is an exception that is typically a user error or a problem that cannot be foreseen by the programmer. For example, if a file is to be opened, but the file cannot be found, an exception occurs. These exceptions cannot simply be ignored at the time of compilation.
- Runtime exceptions: A runtime exception is an exception that occurs that probably could have been avoided by the programmer. As opposed to checked exceptions, runtime exceptions are ignored at the time of compilation.
- Errors: These are not exceptions at all, but problems that arise beyond the control of the user or the programmer. Errors are typically ignored in your code because you can rarely do anything about an error. For example, if a stack overflow occurs, an error will arise. They are also ignored at the time of compilation.

All exception classes are subtypes of the java.lang.Exception class. The exception class is a subclass of the Throwable class. Other than the exception class there is another subclass called Error which is derived from the Throwable class.

The Exception class has two main subclasses: IOException class and RuntimeException Class.



Exceptions Methods:

Following is the list of important methods available in the Throwable class.

Sr no	Methods	Description
1	Public String getMessage()	Returns a detailed message about the exception that has occurred. This message is initialized in the Throwable constructor.
2	public Throwable getCause()	Returns the cause of the exception as represented by a Throwable object.
3	public String toString()	Returns the name of the class concatenated with the result of getMessage()
4	public void printStackTrace()	Prints the result of toString() along with the stack trace to System.err, the error output stream.
5	public StackTraceElement [] getStackTrace()	Returns an array containing each element on the stack trace. The element at index 0 represents the top of the call stack, and the last element in the array represents the method at the bottom of the call stack.
6	public Throwable fillInStackTrace()	Fills the stack trace of this Throwable object with the current stack trace, adding to any previous information in the stack trace.

Example:

The following is an array is declared with 2 elements. Then the code tries to access the 3rd element of the array which throws an exception.

```
public class ExcepTest{
  public static void main(String args[]){
    try{
      int a[] = new int[2];
      System.out.println("Access element three :" + a[3]);
    }catch(ArrayIndexOutOfBoundsException e){
      System.out.println("Exception thrown  :" + e);
    }
    System.out.println("Out of the block");}}
```

This would produce the following result:

```
Exception thrown  :java.lang.ArrayIndexOutOfBoundsException: 3 Out of the block
```

- The throws/throw Keywords:

If a method does not handle a checked exception, the method must declare it using the throwskeyword. The throws keyword appears at the end of a method's signature. You can throw an exception, either a newly instantiated one or an exception that you just caught, by using the throw keyword. Try to understand the different in throws and throw keywords.

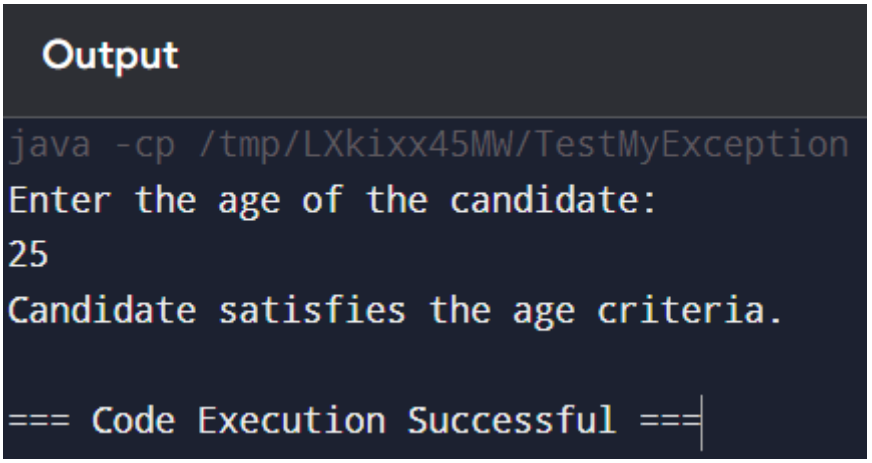
- The finally Keyword

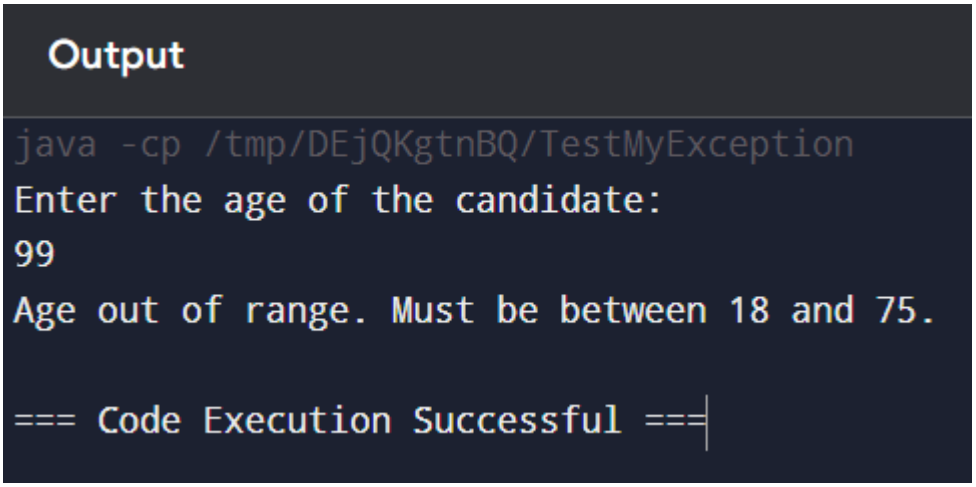
The finally keyword is used to create a block of code that follows a try block. A finally block of code always executes, whether or not an exception has occurred.

Using a finally block allows you to run any cleanup-type statements that you want to execute, no matter what happens in the protected code.

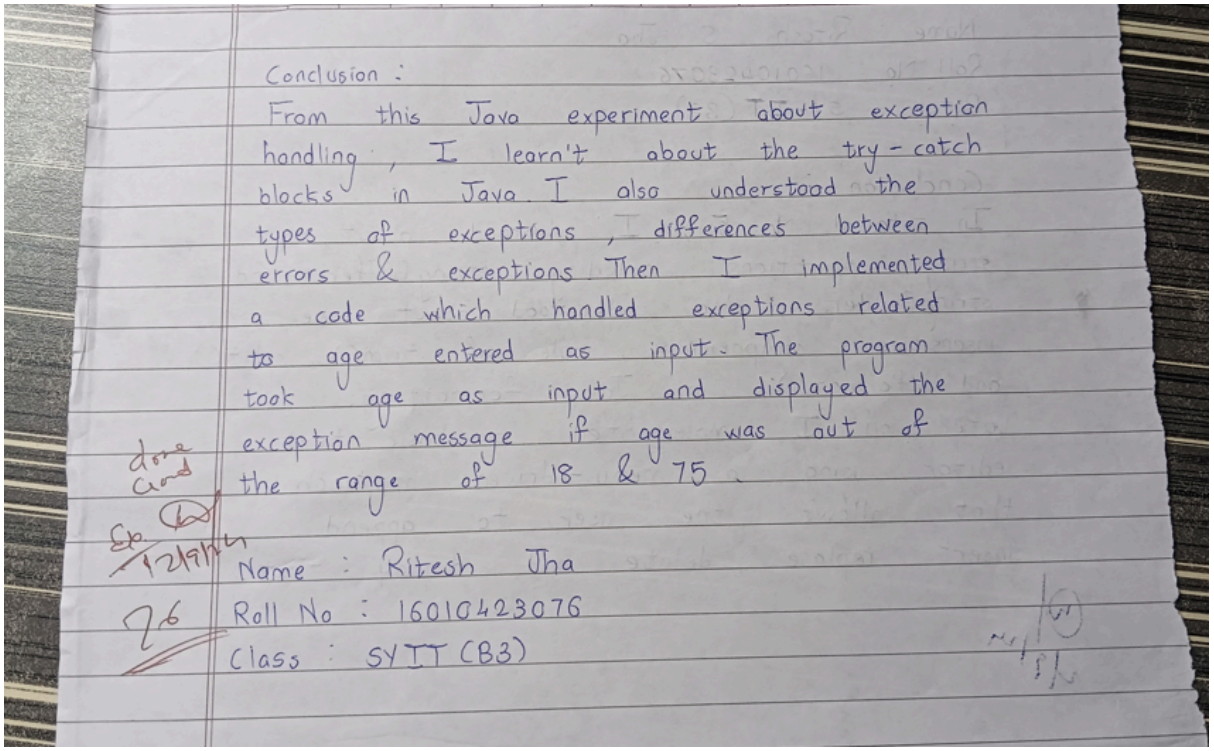
Results: (Program with output)

```
import java.util.Scanner;
class AgeOutOfRangeException extends Exception {
   AgeOutOfRangeException(String message) {
      super(message);
   }
}

public class TestMyException {
   public static void main(String args[]) {
      Scanner scanner = new Scanner(System.in);
      System.out.println("Enter the age of the candidate:");

      try {
      int age = scanner.nextInt();
      int lowerLimit = 18;
      int upperLimit = 75;
      if (age >= lowerLimit && age <= upperLimit) {
      System.out.println("Candidate satisfies the age criteria.");
         }

      else {
         throw new AgeOutOfRangeException("Age out of range. Must be between " +
lowerLimit + " and " + upperLimit + ".");
         }
      }


      catch (AgeOutOfRangeException e) {
      System.out.println(e.getMessage());
      }

      finally {
      scanner.close();
      }
   }
}
```





Conclusion: (Conclusion to be based on the outcomes achieved)



Conclusion :

From this Java experiment about exception handling, I learn't about the try-catch blocks in Java. I also understood the types of exceptions, differences between errors & exceptions. Then I implemented a code which handled exceptions related to age entered as input. The program took age as input and displayed the exception message if age was out of the range of 18 & 75.

done good

12/9/24

Name : Ritesh Jha

Roll No : 16010423076

Class : SY IT (B3)

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

Books/ Journals/ Websites:

1. Herbert Schildt, “Java: The Complete Reference” Tata McGrawHill Publishing Company Limited Tenth Edition, 2017
2. Sachin Malhotra, Saurab Choudhary, “Programming in Java” Oxford University Press Second Edition, 2018 3.
3. D.T. Editorial Services, “Java 8 Programming Black Book” Dream tech Press Edition 2015