Priyansh Salian

Roll No-2003148

Class-C3

Experiment 10 –User Defined Exception

Theory-

User-defined exceptions in Java are used to handle the application-specific error conditions. Here application-specific error conditions are bound to a specific application those are not generic for all the applications.

The guys implemented the Java API, provide us extreme Exception handling mechanism which was generic for all the Java-based applications. In the previous tutorials, we have discussed what is Exceptions and top 10 Exceptions in Java. Now we are going to discuss the importance of user-defined exceptions.

User Defined Exceptions Scenario

The best scenario, when we talk about the user defined exceptions are banking domain applications. In any banking application, the major modules are deposits and withdrawals. When performing a withdrawal from a bank account, it is required to validate the minimum balance in the account. To validate this requirement Java didn’t provide any specific exception class. Hence to handle this

requirement, a user-defined exception MinimumAccountBalance may be created.

The user defined exceptions are generated using the throw statement and handle them as normal exceptions.

If necessary, we can write our own user defined exceptions and throw and handle them. We can create the user defined exceptions by extending the Exception class.

Syntax:

MinimumAccountBalance.java

```
class MinimumAccountBalance extends Exception{

}
```

User defined exceptions class names does not match with already existed Exception classes.

A.

AIM- WAP to accept any integer from the user & if the entered number is not any of 5 or 6 or 7 then create an exception & catch it

Program-

```
import java.util.*;


class MyException extends Exception{
    MyException(String s){
        super(s);
    }
}
class expa{
    static void validate(int num) throws MyException{
        if(num!=5 || num!= 6 || num!=7){
            throw new MyException("Invalid input");
```

```
        }
        else{
            System.out.println("Valid input");
        }
    }
    public static void main(String[] args) {
        try{
            validate(13);
        }
        catch(Exception e){
            System.out.println(e);
        }
    }
}
```

Output-

B.

AIM- WAP to create an exception 'PayOutOfBounds' when the basic pay paid to the Superintendent it is less than 25,000 and greater than 50,000

Program-

```
import java.util.*;
import java.io.*;


class PayOutOfBoundsException extends Exception {
	public PayOutOfBoundsException(String message) {
		super(message);
	}
}


public class TenB {
	static void check_salary(int salary) throws PayOutOfBoundsException {
		if(salary < 25000 || salary > 50000) {
			if(salary < 25000) {
				throw new PayOutOfBoundsException("Salary is less
than 25k!");
			}
```

```
                else {
                        throw new PayOutOfBoundsException("Salary is greater
than 50k!");
                }
        }
        else {
                System.out.print("Salary is in between 25k and 50k.");
        }
    }
  public static void main(String args[]) throws Exception {
    Scanner in = new Scanner(System.in);
        try {
                System.out.print("Enter salary of superintendent: ");
                int salary = in.nextInt();
                check_salary(salary);
        }
        catch(PayOutOfBoundsException e) {
                System.out.print(e);
        }
  }
}
```

Output-


C:\Users\Puru\Desktop\PRIYANSH\College\JAVA LAB WORK>cd "c:\Users\Puru\Desktop\PRIYANSH\College\JAVA LAB WORK\" && javac TenB.java && java TenB
Enter salary of superintendent: 30000
Salary is in between 25k and 50k.
C:\Users\Puru\Desktop\PRIYANSH\College\JAVA LAB WORK>