Java Lab Assessment 8

Aaditya Kumar Muktavarapu
HU21CSEN0100580

6.Create a class Student with attributes roll no, name, age and course. Initialize values through parameterized constructor. If age of student is not in between 15 and 21 then generate user-defined exception "AgeNotWithinRangeException". If name contains numbers or special symbols raise exception "NameNotValidException".
Define the two exception classes.

```
import java.util.regex.Matcher;

import java.util.regex.Pattern;



class AgeNotWithinRangeException extends Exception{

   public AgeNotWithinRangeException(String s) {

       super(s);

   }

}

class NameNotValidException extends Exception{

   public NameNotValidException(String s) {

       super(s);

   }

}

public class Student {

   String name;
```

```
   int age;



   boolean checker(String s) {

       Pattern p = Pattern.compile("[^a-z0-9 ]", Pattern.CASE_INSENSITIVE);

       Matcher m = p.matcher(s);

       boolean res = m.find();

       return res;

   }



   Student(String n, int a){

       try {

           if(!(a>15 && a<21)) {

               throw new AgeNotWithinRangeException("Age not in the required
range");

           }

           else {

               this.age = a;

           }

           if(checker(n)){

               throw new NameNotValidException("Name contains special
characters");

           }

           else {

               this.name=n;

           }
```

```
        }

        catch(AgeNotWithinRangeException ae) {

            System.out.println(ae.getMessage());

        }

        catch(NameNotValidException ne) {

            System.out.println(ne.getMessage());

        }

    }



    public static void main(String args[]) {

        Student s = new Student("Aaditya",20);

        s.checker("Aaditya");

    }

}
```

7.Program to throw a user defined exception for employee details

- If an employee name is a number, a name exception must be thrown.
- If an employee age is greater than 50, an age exception must be thrown

```
import java.util.regex.Matcher;

import java.util.regex.Pattern;



class AgeExceedsLimit extends Exception{

    public AgeExceedsLimit(String s) {

        super(s);
```

```
    }

}

class NameNotValidException extends Exception{

    public NameNotValidException(String s) {

        super(s);

    }

}

public class Employee {

    String ename;

    int eage;



    boolean checker(String s) {

        Pattern p = Pattern.compile("(.)*(\\d)(.)*", Pattern.CASE_INSENSITIVE);

        Matcher m = p.matcher(s);

        boolean res = m.find();

        return res;

    }



    Employee(String en, int ea){

        try {

            if(ea>50) {

                throw new AgeExceedsLimit("Age exceeds the limit");

            }

            else {
```

```
                this.eage = ea;

            }

            if(checker(en)){

                throw new NameNotValidException("Name contains numbers");

            }

            else {

                this.ename=en;

            }

        }

        catch(AgeExceedsLimit AE) {

            System.out.println(AE.getMessage());

        }

        catch(NameNotValidException NE) {

            System.out.println(NE.getMessage());

        }

    }



    public static void main(String args[]) {

        Employee e = new Employee("Aaditya1",35);

        e.checker("Aaditya1");

    }

}
```

Output

8.Program to demonstrate nested exception.

```
public class nestedException {

   static int nums[] = new int[5];

   public static void main(String[] args) {

       try {

           try {

               int num1 = 10;

               int num2 = 0;

               int result = num1 / num2;

           } catch (ArithmeticException e) {

               System.out.println("Error: " + e.getMessage());

               throw new RuntimeException("Inner Exception occurred", e);

           }

           try {

               nums[5]=5;

           }catch(ArrayIndexOutOfBoundsException ae) {

               System.out.println("Error: " + ae.getMessage());

               throw new RuntimeException("Inner Exception occurred", ae);

           }

       } catch (RuntimeException e) {

           System.out.println("Error: " + e.getMessage());

       }
```

```
    }

}
```

Output

Error: / by zero

Error: Inner Exception occurred

9.Create an Account class with data members acco, name,bal. Include methods deposit(), withdraw(). Raise an exception when balance in account is less than 1000.

```
class minBalanceException extends Exception{

    public minBalanceException(String s) {

        super(s);

    }

}



public class Account {

    String accno, name;

    float bal, minBal;



    Account(String n,String ano,float b,float mb){

        this.name = n;

        this.accno = ano;

        this.bal = b;

        this.minBal = mb;
```

```
    }


    void withdraw(float val) {

        try {

            if((bal-val)<minBal) {

                throw new minBalanceException("Balance less than minimum
balance");

            }

            else {

                this.bal-=val;

            }

        }catch(minBalanceException m) {

            System.out.println("Exception: "+m.getMessage());

        }

    }



    void deposit(float val) {

        this.bal+=val;

    }



    public static void main(String args[]) {

        Account a = new Account("Chai","1234",500,500);

        a.withdraw(100);

    }
```

```
}
```

Output

Exception: Balance less than minimum balance

10.Create a Student class with data members Rollno, Name, marks in subjects. Include methods to compute average. Raise an exception if the student has more than 2 backlogs.

```
public class nestedException {
   static int nums[] = new int[5];
   public static void main(String[] args) {
       try {
           try {
               int num1 = 10;
               int num2 = 0;
               int result = num1 / num2;
           } catch (ArithmeticException e) {
               System.out.println("Error: " + e.getMessage());
               throw new RuntimeException("Inner Exception occurred", e);
           }
           try {
               nums[5]=5;
           }catch(ArrayIndexOutOfBoundsException ae) {
               System.out.println("Error: " + ae.getMessage());
               throw new RuntimeException("Inner Exception occurred", ae);
           }
       } catch (RuntimeException e) {
           System.out.println("Error: " + e.getMessage());
       }
   }
}
```

Output
Number of backlogs: 2
53.0