Exception Handling

Solutions

1. **Write Java Program to accept a number from user, if number is zero throw user defined exception "Number is 0" otherwise check whether no. is prime or not(use static keyword).**

My_Exception.java

```
package com.divya.execptions;
public class My_Exception extends Exception {
  public My_Exception(String msg){
     super(msg);
  }
}
```

Zero_Or_Prime.java

```
package com.divya.execptions;

import java.util.Scanner;

public class Zero_Or_Prime {
```

```
    public static void check_exception(int number) throws
My_Exception{
         if(number==0){
              throw(new My_Exception("Number is 0"));
         }
         else{
              check_Prime(number);
         }
    }
     public static void check_Prime(int n){
         int check=0;
         if(n==1){
              System.out.println("The number is not prime");
         }
         else{
              for(int i=2;i<=Math.sqrt(n);i++){
                   if(n%i==0){
                        System.out.println("The number is not
prime");
                        check=1;

                        break;
                   }
```

```
            }
            if(check==0){
            System.out.println("The number is prime");
            }
      }


}

public static void main(String[] args) {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the number");
      int num=sc.nextInt();

            try {
                  check_exception(num);
            } catch (My_Exception e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            }
```

```
        // TODO Auto-generated method stub


    }


}
```

2. **Write 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 "Age Not within Range". If name contains numbers or special symbols, raise exception "Name not Valid".**

Student.java

```
package com.divya.student.execptions;


public class Student {
    int rollno;
    String name;
    int age;
   String course;


    public Student(int rollno, String name, int age, String course) {
```

```
        super();

        this.rollno = rollno;
        this.name = name;
        this.age = age;
        this.course = course;
}

}
```

StudentException.java

```
package com.divya.student.execptions;

public class StudentException extends Exception {

    public StudentException(String msg) {
        super(msg);
        // TODO Auto-generated constructor stub
    }

}
```

StudentClient.java

```
package com.divya.student.execptions;


import java.util.Scanner;


public class StudentClient {

    public static void check_student_exception(Student s)
throws StudentException{

        int i;

        if(s.age>=15 && s.age<=21){

            throw(new StudentException("Age Not within
Range"));

        }

        for(i=0;i<s.name.length();i++){

            char c=s.name.charAt(i);

            if(!Character.isLetter(c)){

                throw(new StudentException("Name not
Valid"));



            }

        }

    }
```

```
public static void main(String[] args) {
     Scanner sc=new Scanner(System.in);
     System.out.println("Enter student rollno");
     int rollno=sc.nextInt();
     System.out.println("Enter student name");
    String name=sc.next();
    System.out.println("Enter student age");
     int age=sc.nextInt();
     System.out.println("Enter student course");
    String course=sc.next();
     Student s1=new Student(rollno,name,age,course);

          try {
               check_student_exception(s1);
          } catch (StudentException e) {
               // TODO Auto-generated catch block
               e.printStackTrace();
          }

     // TODO Auto-generated method stub

}
```

```
}
```

3. **Write a Java Program to find the exception Marks Out of Bounds. Create a class Student. If mark is greater than 100, it must generate user defined exception called Mark Out of Bounds Exception and throw it.**

Student.java

```
package com.student.marks.exception;


public class Student {
     int rollno;
     String name;
     int age;
    String course;
    int marks;
     public Student(int rollno, String name, int age, String course, int marks) {
          super();
          this.rollno = rollno;
          this.name = name;
          this.age = age;
          this.course = course;
```

```
            this.marks = marks;
        }

}
```

MarksOutofBoundsException.java

```
package com.student.marks.exception;

public class MarkOutOfBoundsException extends Exception {
    public MarkOutOfBoundsException(){
        super("Marks valid is less than 100");
    }

}
```

StudentClient.java

```
package com.student.marks.exception;

import java.util.Scanner;

public class StudentClient {
```

```
    public static void main(String[] args) {
         Scanner sc=new Scanner(System.in);
         System.out.println("Enter student rollno");
         int rollno=sc.nextInt();
         System.out.println("Enter student name");
        String name=sc.next();
        System.out.println("Enter student age");
         int age=sc.nextInt();
         System.out.println("Enter student course");
        String course=sc.next();
        try{
        System.out.println("Enter student marks");
         int marks=sc.nextInt();
         if(marks>100) throw new MarkOutOfBoundsException();
         Student s1=new
Student(rollno,name,age,course,marks);
         System.out.println("Details entered successfully");
        }catch (MarkOutOfBoundsException e){
         System.out.println(e);
        }

    }
```

```
}
```

4. **Find out maximum of array elements and check for array limit.**

ArrayLimitException.java

```
package com.divya.array.exception;

public class ArrayLimitException extends Exception {
     public ArrayLimitException(){
          super("Array limit out of bounds");
     }
}
```

ArrayMain.java

```
package com.divya.array.exception;

import java.util.Scanner;

public class ArrayMain {

     public static void inputArray(int[]arr,int x,int n)
throws ArrayLimitException{
```

```
        int i;
        for(i=0;i<n;i++){

            if((arr[i])>=x){
                throw (new ArrayLimitException());
            }
        }



}
static int max_Array(int[]arr){
       int i;
       int max=arr[0];
       for(i=1;i<arr.length;i++){
        if(arr[i]>max){
             max=arr[i];
        }
       }
       return max;
```

```
    }

    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int i=0;
        int max;

        System.out.println("enter the limit");
        int x=sc.nextInt();
        int n;
        System.out.println("enter the size");
        n=sc.nextInt();
        int[]arr = new int[n];

        System.out.println("enter the elements");
        for(i=0;i<n;i++){
            arr[i]=sc.nextInt();
        }
        try {
            inputArray(arr,x,n);
            System.out.println("elements inserted
successfully");
            max=max_Array(arr);
```

```
            System.out.println("The maximum is "+max);
        } catch (ArrayLimitException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }


        // TODO Auto-generated method stub

    }

}
```

5. **Write a class Account with acc_no, name and balance. Initialize values through parameterized constructor. If balance is between 1000 and 5000, generate user defined exception "Balance within the range". If name contains digits, raise exception "Name not Valid".**

Account.java

```
package com.divya.accountExceptions;

public class Account {
    private int acc_no;
```

```
    private String name;
    private int balance;
    public Account(int acc_no, String name, int balance) {
        super();
        this.acc_no = acc_no;
        this.name = name;
        this.balance = balance;
    }

}
```

BalanceException.java

```
package com.divya.accountExceptions;

public class BalanceException extends Exception{
    public BalanceException(){
        super("Balance within range");
    }

}
```

NameException.java

```
package com.divya.accountExceptions;
```

```
public class NameException extends Exception {
     public NameException(){
          super("Name not valid");
     }

}
```

AccountClient.java

```
package com.divya.accountExceptions;

import java.util.Scanner;

public class AccountClient {

     public static void main(String[] args) {
          Scanner sc=new Scanner(System.in);
          System.out.println("enter account number");
          int acc_no=sc.nextInt();
          System.out.println("enter name");
          try {
          String name=sc.next();
          int i;
```

```
            for(i=0;i<name.length();i++){
                 char c=name.charAt(i);
                 if(Character.isDigit(c)) throw new
NameException();
            }
            System.out.println("enter balance");
            int balance=sc.nextInt();
            if(balance>=1000 && balance<=5000) throw new
BalanceException();
            Account ac=new Account(acc_no,name,balance);
            System.out.println("Details entered successfully");
                 } catch (BalanceException|NameException e) {
                      // TODO Auto-generated catch block
                      e.printStackTrace();
                 }

            }
            // TODO Auto-generated method stub

      }
```

6**. Class ExceptionDemo throws following exception depending upon following condition:**

- ✓ **Take any integer from the keyboard.**

- ✓ **If the integer is between 0 and 5, exception of type "Small Number" is thrown.**
- ✓ **If the number is between 5 and 10,"Proper Number" is thrown.**
- ✓ **If the number is greater than 10,"Greater Number" is thrown.**
- ✓ **Also find the factorial of that number(using static keyword).**

ExceptionDemo.java

```
package com.divya.exceptiondemo;

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

}
```

ExceptionDemoClient.java

```
package com.divya.exceptiondemo;

import java.util.Scanner;

public class ExceptionDemoClient {
```

```
static int fact(int n){
     if(n==0||n==1){
          return 1;
     }
     else{
     return n*fact(n-1);
     }
}
     public static void main(String[] args) {
          Scanner sc=new Scanner(System.in);
          System.out.println("enter the number");
          try {
          int x=sc.nextInt();
          if(x>=0 && x<=5) throw new ExceptionDemo("Proper
Number");
          else if(x>10) throw new ExceptionDemo("Greater
Number");
          else{
               int fac=fact(x);
               System.out.println("The factorial of x is
"+fac);
          }
               } catch (ExceptionDemo e) {
```

```
                    // TODO Auto-generated catch block
                    e.printStackTrace();
               }
          // TODO Auto-generated method stub

     }

}
```

7. Class ExceptionDemo throws the following exception depending upon the following conditions. Take any string from the keyboard.

- ✓ **If string length is between 0 and 5, then Exception "too small string" is thrown.**
- ✓ **If string length is between 5 and 10, then exception to type "Perfect String" is thrown.**

ExceptionDemo.java

```
package com.divya.newExceptionDemo;


public class ExceptionDemo extends Exception {


          public ExceptionDemo(String message){
               super(message);
          }
```

```
}
```

ExceptionDemoClient.java

```
package com.divya.newExceptionDemo;

import java.util.Scanner;

public class ExceptionDemoClient {

     public static void main(String[] args) {
          // TODO Auto-generated method stub
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter the string");
    try {
    String name=sc.next();
    if(name.length()>0 && name.length()<=5)

               throw new ExceptionDemo("Too small string");
    else if (name.length()>5 && name.length()<10)
     throw new ExceptionDemo("Perfect String");
```

```
    else{
        System.out.println("The string is perfectly
entered");
    }

        } catch (ExceptionDemo e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

    }

}
```

8. **Program which accepts string from user. If string contains word "India", raise an exception.**

MyException.java

```
package com.divya.myexceptions;

public class MyException extends Exception{
    public MyException(){
        super("String contains word India");
```

```
    }

}
```

MyExceptionClient.java

```
package com.divya.myexceptions;

import java.util.Scanner;

public class MyExceptionClient {

    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("enter your string");
        String word="India";

        String sentence=sc.nextLine();
        try {
        if(sentence.contains(word))throw new MyException()
;
        else{
        System.out.println("String entered successfully");
```

```
        }


            } catch (MyException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        // TODO Auto-generated method stub

    }

}
```

9. **Write a Java program to find sum of integer from command line argument and count invalid integers through command line(Use Exception Handling).**

CommandExceptions.java

```
public class CommandExceptions {
      public static void main(String[] args) {
            int sum = 0;
            int count=0;
            int num=0;
```

```
            for (int i = 0; i < args.length; i++) {
             try{
             num=Integer.parseInt(args[i]);
             }catch(NumberFormatException e){
                  count++;
                  System.out.println("The error is Invalid
number");
             }
                sum = sum + num;
            }
            System.out.println("The sum of the arguments
passed is " + sum);
            System.out.println("the sum of invalid numbers
is "+count);
         }


}
```

In commandPrompt

```
C:\Users\jmuralee\Desktop\Downloads\java_workspace\divya_exc
eptions\bin>java Com
mandExceptions 2 3 4
The sum of the arguments passed is 9
```

```
the sum of invalid numbers is 0


C:\Users\jmuralee\Desktop\Downloads\java_workspace\divya_exc
eptions\bin>java Com

mandExceptions 25 d 6

The error is Invalid number.To be exact
java.lang.NumberFormatException: For inp

ut string: "d"

The sum of the arguments passed is 56

the sum of invalid numbers is 1
```

10. **Write a program in Java to check if entered data is alphabet or digit. If it is alphabet, then print if it is capital or small case. If digit is entered throw user defined exception "digit not allowed".**

DigitException.java

```
package com.digit.exception;


public class DigitException extends Exception {
     public DigitException(){
          super("digit not allowed");
     }
```

```
}
```

DigitClient.java

```
package com.digit.exception;

import java.util.Scanner;

public class DigitClient {

     public static void main(String[] args) {
          Scanner sc=new Scanner(System.in);
          System.out.println("enter your string");
          String check=sc.next();
          int i;
          int lcount=0,ucount=0;
          try {
          for(i=0;i<check.length();i++){
          char c=check.charAt(i);
          if(Character.isLowerCase(c)){
               lcount++;
          }
          else if(Character.isUpperCase(c)){
```

```
                ucount++;
            }
            else if(!Character.isLetter(c)) throw new
DigitException();
            }
                } catch (DigitException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
            if(lcount!=0 && ucount==0){
                System.out.println("The letters are small
letters");
            }
            if(ucount!=0 && lcount==0){
                System.out.println("The letters are capital
letters");
            }
            else if(lcount!=0 && ucount!=0){
                System.out.println("The letters contain both
capital and small letters");
            }
}
    }
```

11. **Write program demonstrating nested try blocks and multiple catch statements and using throw and throws and finally keywords.**

ExceptionEx.java

```
package com.exceptions;


import java.util.Scanner;


public class ExceptionEx {
     void funA() throws
ArithmeticException,NullPointerException{
          int x=100;
          int y=100/0;
          System.out.println(y);
     }
     public static void main(String[] args) {
          ExceptionEx ex=new ExceptionEx();
          try{
          try{
               System.out.println("To show arithmetic
exception");
```

```
            int x=100/0;


        }catch(ArithmeticException e){
            System.out.println("Arithmetic Exception");
        }
        try{
            int a[]=new int[6];
            a[7]=10;
        }catch(ArrayIndexOutOfBoundsException e){
            System.out.println("Array Index out of bounds
Exception");
        }
        try{
            ex.funA();


        }catch(ArithmeticException|NullPointerException e){
            System.out.println("Either arithmetic or null
pointer exception.to be exact"+e);
        }
        }catch(Exception e){
            System.out.println("Handling exception");
        }
        finally{
```

```
                System.out.println("to perform the cleanup
activities");
          }



     }


}
```