

CSE1007

JAVA PROGRAMMING

LAB ASSIGNMENT-3

Name: Yogeswari Sahu

Registration Number: 18BCE0928

Slot: L29+L30

Faculty: KUMAR P.J

QUESTION :

Write a program(s) in Java that illustrates the following concepts. The concepts can be applied to any practical scenario such as student administration system or software which maintains various information about students and provides various services to the users. The other scenarios may include retail business management software, banking systems, railway reservation system, online shopping applications, environmental monitoring system etc.

- Exception Handling
- Thread creation
- Thread synchronization

SOLUTION:

PROGRAM 1:

In a real life scenario of a filling a form in VIT University, Read the Register Number and Mobile Number of a student. If the Register Number does not contain exactly 9 characters or if the Mobile Number does not contain exactly 10 characters, throw an `IllegalArgumentException`. If the Mobile Number contains any character other than a digit, raise a `NumberFormatException`. If the Register Number contains any character

other than digits and alphabets, throw a NoSuchElementException. If they are valid, print the message 'valid' else 'invalid'

Concepts used:

- Throwing Exceptions
- Usage of patterns in strings

Code:

```
package JavaDA3;
import java.util.NoSuchElementException;
import java.util.Scanner;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class DA3Prog1 {
    static void validate(String r, String n){
        if(r.length() != 9){
            System.out.println("Invalid");
            throw new IllegalArgumentException("Register Number does
not contain exactly 9 characters");
        }
        if(n.length() != 10){
            System.out.println("Invalid");
            throw new IllegalArgumentException("Mobile Number does not
contain exactly 10 characters");
        }
        String pattern = "[6|7|8|9]{1}\\d{9}";
        Pattern a = Pattern.compile(pattern);
        Matcher m1 = a.matcher(n);
        if(!m1.find()){
            throw new NumberFormatException("Mobile Number cannot
contain any character other than a digit");
        }
        String pattern2 = "[1-9]{2}[A-Z]{3}[0-9]{4}$";
        Pattern b = Pattern.compile(pattern2);
        Matcher m2 = b.matcher(r);
        if(!m2.find()){
            throw new NoSuchElementException("Registration Number
cannot contain any character other than digits and alphabets");
        }
    }

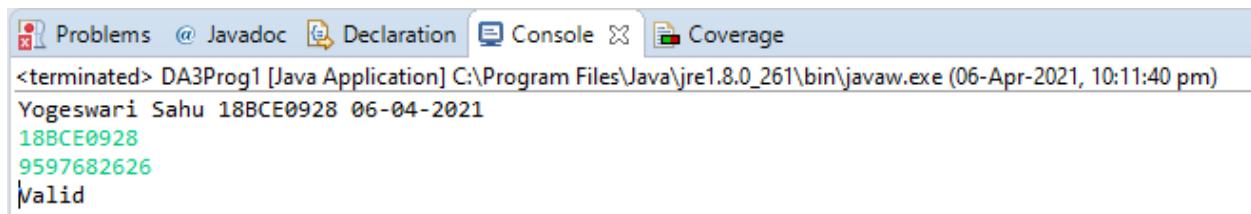
    public static void main(String[] args) {
        System.out.println("Yogeswari Sahu 18BCE0928 06-04-2021");
        Scanner sc = new Scanner(System.in);
    }
}
```

```

        String reg = sc.nextLine();
        String no = sc.nextLine();
        sc.close();
        validate(reg, no);
        System.out.println("Valid");
    }
}

```

Output:



Problems Javadoc Declaration Console Coverage

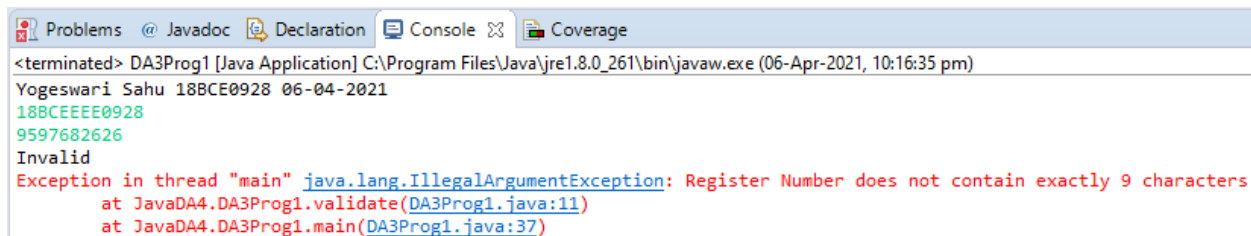
<terminated> DA3Prog1 [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (06-Apr-2021, 10:11:40 pm)

Yogeswari Sahu 18BCE0928 06-04-2021

18BCE0928

9597682626

Valid



Problems Javadoc Declaration Console Coverage

<terminated> DA3Prog1 [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (06-Apr-2021, 10:16:35 pm)

Yogeswari Sahu 18BCE0928 06-04-2021

18BCE0928

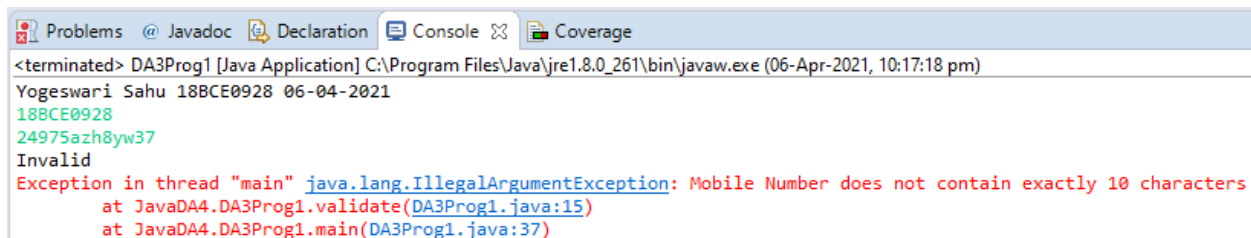
9597682626

Invalid

Exception in thread "main" java.lang.IllegalArgumentException: Register Number does not contain exactly 9 characters

at JavaDA4.DA3Prog1.validate(DA3Prog1.java:11)

at JavaDA4.DA3Prog1.main(DA3Prog1.java:37)



Problems Javadoc Declaration Console Coverage

<terminated> DA3Prog1 [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (06-Apr-2021, 10:17:18 pm)

Yogeswari Sahu 18BCE0928 06-04-2021

18BCE0928

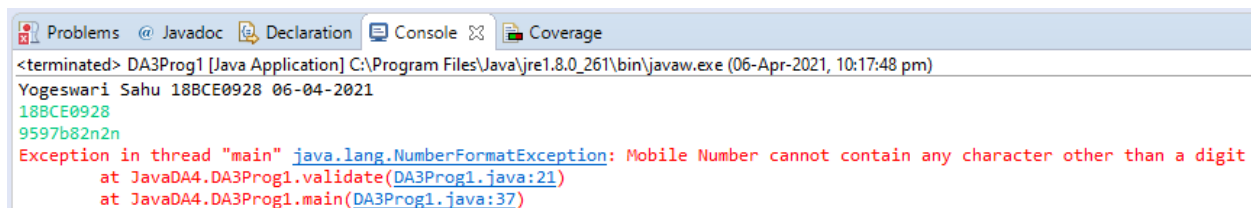
24975azh8yw37

Invalid

Exception in thread "main" java.lang.IllegalArgumentException: Mobile Number does not contain exactly 10 characters

at JavaDA4.DA3Prog1.validate(DA3Prog1.java:15)

at JavaDA4.DA3Prog1.main(DA3Prog1.java:37)



Problems Javadoc Declaration Console Coverage

<terminated> DA3Prog1 [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (06-Apr-2021, 10:17:48 pm)

Yogeswari Sahu 18BCE0928 06-04-2021

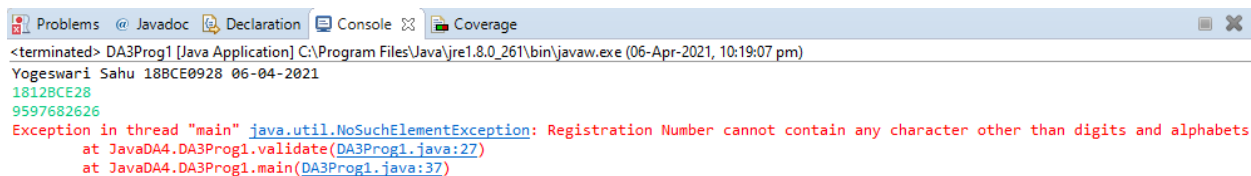
18BCE0928

9597b82n2n

Exception in thread "main" java.lang.NumberFormatException: Mobile Number cannot contain any character other than a digit

at JavaDA4.DA3Prog1.validate(DA3Prog1.java:21)

at JavaDA4.DA3Prog1.main(DA3Prog1.java:37)



Problems Javadoc Declaration Console Coverage

<terminated> DA3Prog1 [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (06-Apr-2021, 10:19:07 pm)

Yogeswari Sahu 18BCE0928 06-04-2021

1812BCE28

9597682626

Exception in thread "main" java.util.NoSuchElementException: Registration Number cannot contain any character other than digits and alphabets

at JavaDA4.DA3Prog1.validate(DA3Prog1.java:27)

at JavaDA4.DA3Prog1.main(DA3Prog1.java:37)

PROGRAM 2:

Let's assume we are playing a game. In this game ,A bag contains balls of 4 different colors- red, green, blue and yellow. Simulate picking up a ball at random for ten times. If the same colored ball is picked more than thrice, throw SameColorBallException and proceed with the simulation once again. After 10 valid picks, print the number of balls chosen from each of these colors.

Concepts used:

- Exception Handling
- Usage of throws and throw
- Printing the exception

Code:

```
package JavaDA3;
import java.util.Random;
class java{
    void choose() throws sameballcolourexception
    {
        Random t = new Random();
        int i,red=0,green=0,blue=0,yellow=0;
        for(i=1;i<=10;i++)
        {
            int ball=t.nextInt(4);
            if(ball==0)
                red++;
            else if(ball==1)
                green++;
            else if(ball==2)
                blue++;
            else if(ball==3)
                yellow++;
        }
        System.out.println("\nBalls picked: \nred:" +red+ "\ngreen:"
+green+ "\nblue:" +blue+ "\nyellow:"+yellow);
        if(red>3 || green>3 || blue>3 || yellow>3)
            throw new
sameballcolourexception("sameballcolourexception");
    }
}

class sameballcolourexception extends Exception
{
    sameballcolourexception(String s)
```

```

        {
            super(s);
        }
    }

    public class DA3Prog2 {

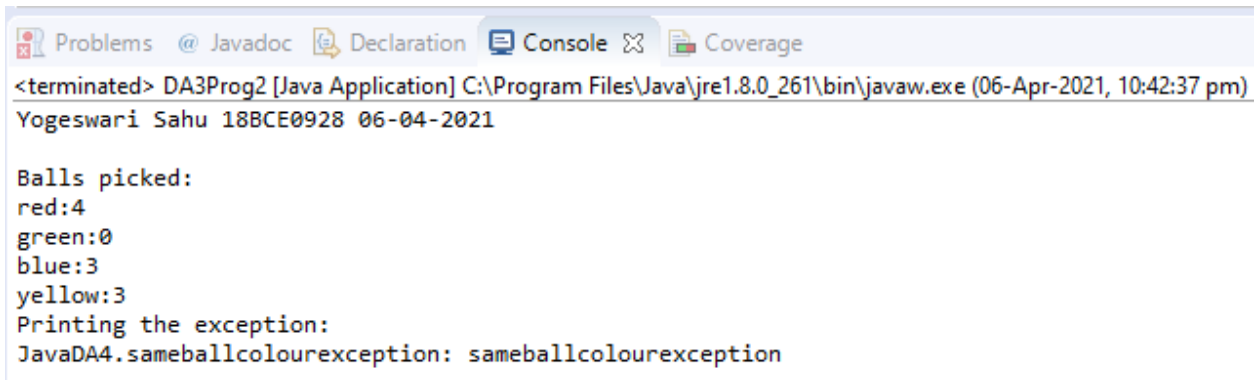
        public static void main(String[] args) {
            System.out.println("Yogeswari Sahu 18BCE0928 06-04-2021");
            try
            {
                java ch=new java();
                ch.choose();
            }
            catch(Exception e)
            {
                System.out.println("Printing the exception:");
                System.out.println(e);
            }

        }

    }
}

```

Output:



```

<terminated> DA3Prog2 [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (06-Apr-2021, 10:42:37 pm)
Yogeswari Sahu 18BCE0928 06-04-2021

Balls picked:
red:4
green:0
blue:3
yellow:3
Printing the exception:
JavaDA4.sameballcolourexception: sameballcolourexception

```

PROGRAM 3:

In a university elections are being conducted. The elections are taken place in the following manner: Let's say Three students A, B and C of B.Tech- II year contest for the election. With the total strength of 50 students in II year, simulate the vote casting by generating 100 random numbers (1 for student A, 2 for B and 3 for C) and store them in an array. Create four

threads to equally share the task of counting the number of votes cast for all the three candidates. Use synchronized method or synchronized block to update the three count variables. The main thread should receive the final vote count for all three contestants and hence decide the PR based on the values received

Concepts used:

- Exception Handling
- Thread creation using Thread class
- Thread synchronization

Code:

DA3Prog3.java

```
package JavaDA3;
import JavaDA3.vote;
import JavaDA3.count;
import java.util.Vector;
public class DA3Prog3 {
    public static void main(String[] args) {
        System.out.println("Yogeswari Sahu 18BCE0928 06-04-2021");
        Vector votevec = new Vector(50);
        vote a = new vote(1, votevec);
        a.start();
        vote b = new vote(2, votevec);
        b.start();
        vote c = new vote(3, votevec);
        c.start();
        try{
            a.join();
            b.join();
            c.join();
            System.out.println("Voting has ended!");
        }catch(Exception e){
            System.out.println(e);
        }
        count ac = new count(1, votevec);
        count bc = new count(2, votevec);
        count cc = new count(3, votevec);
        ac.start();
        bc.start();
        cc.start();
        try{
```

```

        ac.join();
        bc.join();
        cc.join();
        System.out.println("Counting has ended!");
    }catch(Exception e){
        System.out.println(e);
    }
    int av = ac.count;
    int bv = bc.count;
    int cv = cc.count;
    System.out.println("elections.Vote Vector:" + "\n" + votevec);
    System.out.println(av + " votes for A");
    System.out.println(bv + " votes for B");
    System.out.println(cv + " votes for C");
    if(av >= bv && av >= cv){
        if(av == bv || av == cv)
            System.out.println("Tie in elections!");
        else
            System.out.println("A has won the elections!");
    }
    else if(bv >= av && bv >= cv){
        if(av == bv || bv == cv)
            System.out.println("Tie in elections!");
        else
            System.out.println("B has won the elections!");
    }
    else if(cv >= av && cv >= bv){
        if(cv == bv || cv == av)
            System.out.println("Tie in elections!");
        else
            System.out.println("C has won the elections!");
    }
}
}

```

vote.java

```

package JavaDA3;

import java.util.Random;
import java.util.Vector;

public class vote extends Thread{
    Random rand = new Random();
    int max = 750;
    int min = 100;
    int v, s;
    Vector vec;
}

```

```

public vote(int v, Vector vec)
{
    this.v = v;
    this.vec = vec;
}
public void run() {
    try
    {
        while(vec.size() < 50) {
            System.out.println(this.getName() + " is Voting");
            vec.add(v);
            s = rand.nextInt((max - min) + 1) + min;
            System.out.println(this.getName() + " is sleeping for " + s);
            Thread.sleep(s);
        }
    }
    catch(InterruptedException e)
    {
        System.out.println("Voting Exception: " + e);
    }
}
}

```

count.java

```

package JavaDA3;
import java.util.Vector;

public class count extends Thread{
    Vector vec;
    int k, i;
    public int count = 0;
    public count(int k, Vector vec){
        this.k = k;
        this.vec = vec;
    }

    public void run(){
        for(i = 0; i < vec.capacity(); i++){
            if(vec.elementAt(i).equals(k))
                count++;
        }
    }
}

```

Output:

Problems @ Javadoc Declaration Console Coverage

<terminated> DA3Prog3 [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (06-Apr-2021, 11:15:18 pm)

Yogeswari Sahu 18BCE0928 06-04-2021

Thread-0 is Voting
Thread-2 is Voting
Thread-1 is Voting
Thread-2 is sleeping for 569
Thread-0 is sleeping for 215
Thread-1 is sleeping for 511
Thread-0 is Voting
Thread-0 is sleeping for 656
Thread-1 is Voting
Thread-1 is sleeping for 553
Thread-2 is Voting
Thread-2 is sleeping for 706
Thread-0 is Voting
Thread-0 is sleeping for 626
Thread-1 is Voting
Thread-1 is sleeping for 383
Thread-2 is Voting
Thread-2 is sleeping for 567
Thread-1 is Voting
Thread-1 is sleeping for 500

Thread-0 is Voting
Thread-0 is sleeping for 108
Thread-0 is Voting
Thread-0 is sleeping for 539
Thread-2 is Voting
Thread-2 is sleeping for 417
Thread-1 is Voting
Thread-1 is sleeping for 717
Thread-0 is Voting
Thread-0 is sleeping for 106
Thread-0 is Voting
Thread-0 is sleeping for 730
Thread-2 is Voting
Thread-2 is sleeping for 612
Thread-1 is Voting
Thread-1 is sleeping for 225
Thread-2 is Voting
Thread-2 is sleeping for 567
Thread-1 is Voting
Thread-1 is sleeping for 565
Thread-0 is Voting

Thread-0 is sleeping for 606
Thread-2 is Voting
Thread-2 is sleeping for 492
Thread-1 is Voting
Thread-1 is sleeping for 247
Thread-0 is Voting
Thread-0 is sleeping for 705
Thread-1 is Voting
Thread-1 is sleeping for 249
Thread-2 is Voting
Thread-2 is sleeping for 420
Thread-1 is Voting
Thread-1 is sleeping for 147
Thread-1 is Voting
Thread-1 is sleeping for 623
Thread-0 is Voting
Thread-0 is sleeping for 716
Thread-2 is Voting
Thread-2 is sleeping for 126

Thread-1 is Voting
Thread-1 is sleeping for 516
Thread-0 is Voting
Thread-0 is sleeping for 638
Thread-2 is Voting
Thread-2 is sleeping for 382
Thread-1 is Voting
Thread-1 is sleeping for 305
Thread-1 is Voting
Thread-1 is sleeping for 630
Thread-2 is Voting
Thread-2 is sleeping for 699
Thread-0 is Voting
Thread-0 is sleeping for 598
Thread-1 is Voting
Thread-1 is sleeping for 454
Thread-0 is Voting
Thread-0 is sleeping for 547
Thread-2 is Voting
Thread-2 is sleeping for 709
Thread-1 is Voting
Thread-1 is sleeping for 359
Thread-0 is Voting
Thread-0 is sleeping for 551
Thread-2 is Voting
Thread-2 is sleeping for 535
Thread-1 is Voting

```

Thread-1 is sleeping for 425
Thread-0 is Voting
Thread-0 is sleeping for 266
Thread-1 is Voting
Thread-1 is sleeping for 533
Thread-2 is Voting
Thread-2 is sleeping for 316
Thread-0 is Voting
Thread-0 is sleeping for 670
Thread-2 is Voting
Thread-2 is sleeping for 406
Voting has ended!
Counting has ended!
elections.Vote Vector:
[1, 3, 2, 1, 2, 3, 1, 2, 3, 2, 1, 1, 3, 2, 1, 1, 3, 2, 3, 2, 1, 3, 2, 1, 2, 3, 2,
2, 1, 3, 3, 2, 1, 3, 2, 2, 3, 1, 2, 1, 3, 2, 1, 3, 2, 1, 2, 3, 1, 3]
16 votes for A
18 votes for B
16 votes for C
B has won the elections!

```

PROGRAM 4:

Demonstrate multithreading and synchronization by creating two threads, one for printing the odd numbers and the other for printing even numbers with in a given range

Concepts used:

- Exception Handling
- Thread creation by implementing Runnable interface
- Thread synchronization

Code:

```

package JavaDA3;

class java1 implements Runnable {
    public int number1 = 10;
    static int number2 = 2;
    int div;
    static Object Lock = new Object();
    java1(int div) {
        this.div = div;
    }

    @Override
    public void run() {
        while (number2 < number1) {
            synchronized (Lock) {

```

```

        while (number2 % 2 != div) {
            try {
                lock.wait();
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        System.out.println(Thread.currentThread().getName() +
" " + number2);

        number2++;
        lock.notifyAll();
    }
}

public class DA3Prog4 {
    public static void main(String[] args) {
        System.out.println("Yogeswari Sahu 18BCE0928 06-04-2021");
        java1 or=new java1(1);
        java1 er=new java1(0);
        Thread t1=new Thread(or,"Odd");
        Thread t2=new Thread(er,"Even");
        t1.start();
        t2.start();
    }
}

```