Lec-1

Exception Handling

PAGE NO.
DATE :

- Exception :- म्हणजे program चा Normal Flow मध्ये काहीतरी होऊन program थांबतो.
- Program चा formal Flow तुटतो म्हणजे Exception येतो.
- Exception Handling म्हणजे Normal Flow मध्ये आलेले Exception हे Normal Termination देण्यासाठी Exception Handling असते.
- Exception Handling हा program चा flow Abrupt Brake न होण्यासाठी वापरतात.

① Class ExceptionDemo{

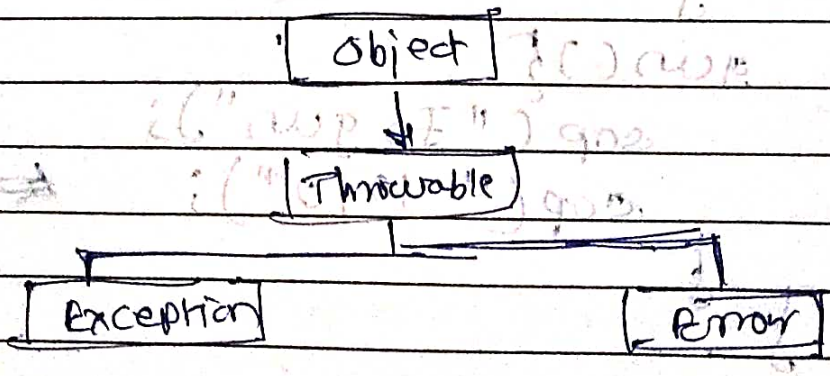
```
    p s v m(String[] args){
        fun();
        sop("In main");
    }
    static void fun(){
        sop("In fun");
        gun();
    }
    gun(){
        sop("In gun");
        sop("10/0");
    }
}
```

⇐ Arithmatic Exception

⇒ Arithmatic Exception

```
Class ExceptionDemo {
    p s v m(String[] args) {
        sop("In main");
        Exception obj = null;   <= NullPointer Exception
        obj.fun();
    }
    void fun() {
        sop("In fun");
        gun();
    }
    void gun() {
        sop("In gun");
    }
}
```

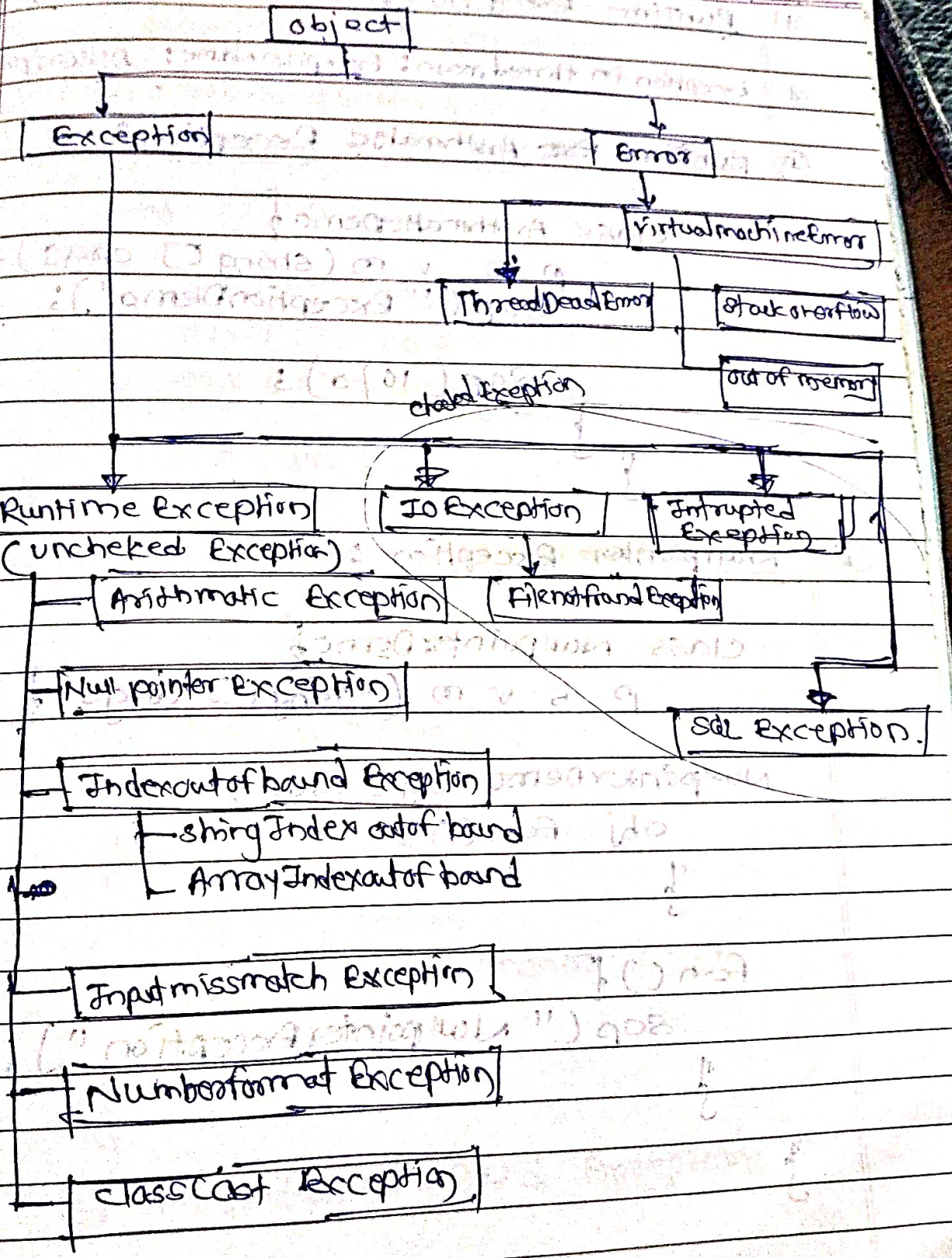
⇒ Null pointer exception हे compile time ला येत नाही ते run time ला येते.



Lec-2

(Runtime Exception) Runtime Exception

Exception handling



RunTime Exception (Unchecked Exception) :-

Exception in thread main: ExceptionName: Description

① ~~RunTime Exc~~ Arithmatic Exception :-

```
class ArithmaticDemo{
    p s v m(String[] args){
    sop("ExceptionDemo");
        sop(10/0);
    }
}
```

② Nullpointer Exception :-

```
class NullpointerDemo{
    p s v m.(string[] args){
NullpointerDemo obj = Null;
    obj.fun();
    }

fun(){
    sop("Nullpointer Exception");
    }
}
```

③ IndexoutofbounsException :-

① ArrayIndex out of bound :-

```
class IndexoutDemo {
        p s v m (string[] args) {
    int arr[] = new int [3];

        arr [0] = 10;
        arr [1] = 20;
        arr [2] = 30;

        sop (arr [3]);
} }
```

⇒ ArrayIndexoutofBound Exception.

② StringIndexoutofBound Exception :-

```
class IndexoutDemo {
    p s v m (string[] args) {
        string s = "shashi";
        sop (s.charAt (7));
    }
}
```

⇒ stringIndexoutofBound Exception.

④ Inputmismatch Exception :-

```
import java.util.*;
class InputmismatchDemo {
    p s v m (String [] args) {
        Scanner sc = new Scanner (System.in);
        int a = sc.nextInt();
        sop (a);
    }
}
```

⇒ Run केल्यानंतर input जर integer दिले तर Exception येत नाही

⇒ int सोडून जर string or दुसरे काही दिले तर Inputmissmatch Exception येते.

⑤ Numberformat Exception :-

```
import java.io.*;
class NumberformatDemo {
    p s v m (String [] args) throws IOException
    {
        BufferedReader br = new BufferedReader (new
                InputStreamReader (System.in));
        int a = Integer.parseInt (br.readline());
        sop (a);
    }
}
```

⇒ Input Diffent प्रकारचा Exception येतो

Lec-3

PAGE NO.
DATE :

⑥ **ClassCast Exception :-**

```
class ClassCastDemo{
    p s v m(String[] args){

        Object O = new Integer(10);
        String s = (String)O;
    }
}
```

CompileTime Exception (Checked Exception) :-

① **IOException :-**

```
import java.io.*;
class IOExceptionDemo{
    p s v m(String[] args){
BufferedReader br = new BufferedReader
                    (new InputStreamReader(System.in));
        int a = Integer.parseInt(br.readLine());
    }
}
```

⇒ IOException occurs because Exception not thrown or declared.

* सगळे RuntimeException हे Throwable class Handle करतो.

② FileNotFoundException :-

```
class FileNotDemo {
    p s v m(String[] args) {
        File f = new File("D:\\web.txt");
        FileInputStream fis = new FileInputStream(f);
    }
}
```

⇒ FileNotFoundException.

③ Interrupted Exception :-

```
class InterruptedDemo {
    p s v m(String[] args) {
        Thread t = new Thread();
        t.start();
        sop("D:\\web");
        t.sleep(5000);
        sop("Biencaps");
    }
}
```

⇒ InterruptedException

PAGE NO.
DATE :

TryCatch of code :-

without TryCatch :-

```
class TryCatchDemo{
    p s v m (String[] args){
        sop ("Before Exception occurs");

    TryCatchDemo obj = null;
        obj.fun();
        sop(" after Exception occurs");
    }

    void fun(){
        sop("In fun()");
    }
```

Lect-4

PAGE NO.
DATE :

TryCatch

checked Exception throws केले जातात
Unchecked Exception throws होते पण त्याचा काही उपयोग नसतो कारण Runtime Exception throws केलेले Consider केले जात नाही.

```
try {
    Risky Code
}
catch (Exception) {
    Handling code
}
```

①
```
class TryRuntimeDemo {
    p s v m (String[] args) {

    try {
        sop(10/0);
    }
    catch (ArithmeticException obj) {
        sop("Exception Handled");
    }

    sop("After try catch");
    }
}
```

② class TryRuntimeException {
p s v m (String[] args) {
int x=10; try {
TryRuntimeDemo obj=new TryRuntimeDemo();
~~sop(x);~~ sop(obj.x);
} catch (NullPointerException obj) {
sop("Exception Handled");
}
sop("After try catch");
}}

for Exception Handling :-

① In above program

TryRuntimeDemo obj=null;

② In above program

change ⇒

catch (ArithmeticException obj)

Default Exception Handler


```
# import java.io.*

class checkedDemo {
    p s v m (String[] args) throws IOException {
BufferedReader br = new BufferedReader
        (new InputStreamReader(System.in));

    sop( br.readLine());

      br.close ();
      try {
          sop (br.readLine());
      }
      catch (IOException obj) {
            sop(" Buffered closed");
      }
  }
}
```

जब IO Exception throws नहीं करते तब try catch से IOException Handled करना पड़ता

e.g.

```
class CheckedDemo {
  p s v m (String[] args) {
BufferedReader br = new BufferedReader(
        InputStreamReader (System.in));

   try {
        sop (br.readLine());
   }
```

PAGE NO.
DATE:

```
catch (IOException obj) {
    sop("IO Exception Handled");
}
sop("End of program");
}
}
```

class ExceptionDemo {

p s v m(String[] args)

Exception Handling :-

* Try Catch :-

(i) class ExceptionDemo {
    p s v m (String[] args) {
        sop ("Before Exception");
        try {
            sop (10/0);
        }
        catch (Exception e) {
            sop ("Handling code");
        }
        sop ("After Exception");
    }
}

② Class ExceptionDemo
p s v m (String[] args) {
sop ("Befor Exception");

try {
sop (10/0);
} catch (ArithmeticException e) {

sop("Exception Handlled");
}

sop ("After Exception");
}
}

③ class Core2web extends RuntimeException {

int a = 20;
}

class ExceptionDemo {

sop("Before Exception");

try {

sop (10/0);
}

```
catch {

catch (core2web e) {

        sop("After Exception");

}  }
```