



Saraswati Vandana

या कुन्देन्दु तुषार हार धवला
या शुभ्र वस्त्रान्विता ।
या वीणा वर दंड मंडितकरा
या श्वेत पद्मासना ॥

या ब्रह्मा अच्युत शंकर प्रभृतिभिः
देवै सदा पूजिता ।
सा मां पातु सरस्वती भगवती
निःश्रेयेश जाङ्घापह ॥

Java Programming (1ET1030406)

Unit-8 : Exception Handling

Prepared By

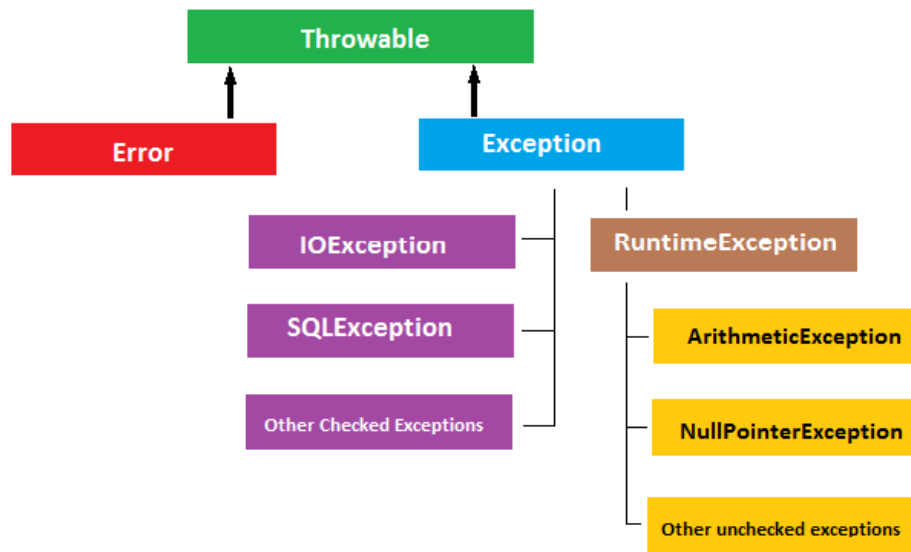
Mr. Mehul S. Patel

Department of Computer Engineering & Information Technology

Content

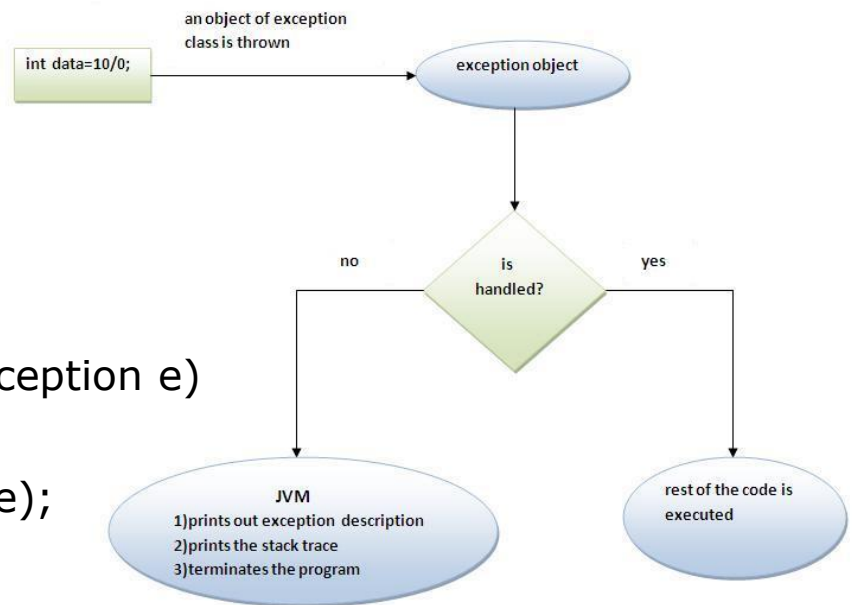
- Exception and Error
- Use of try, catch
- throw, throws and finally
- Built in Exception
- Custom exception
- Throwable Class

Introduction

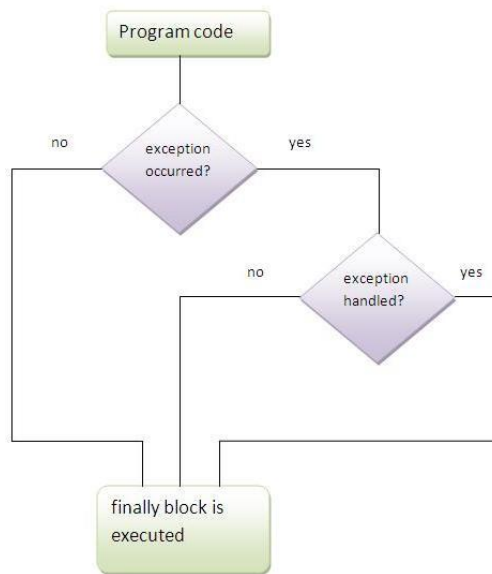


Use of try, catch

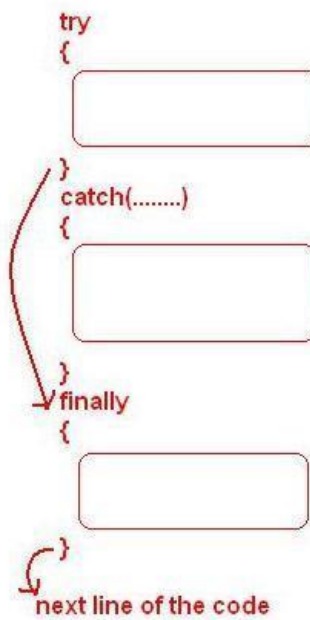
```
try{  
    int data=a/b;  
}  
catch(ArithmeticException e)  
{  
    System.out.println(e);  
}
```



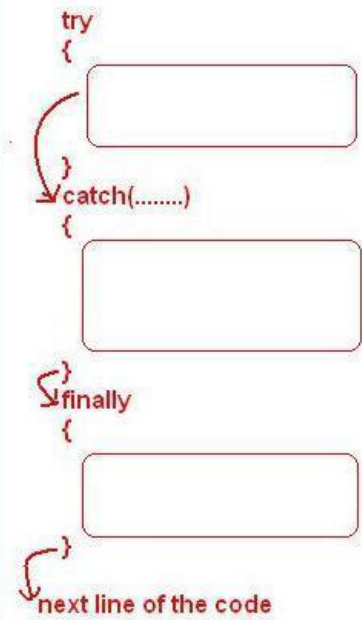
finally



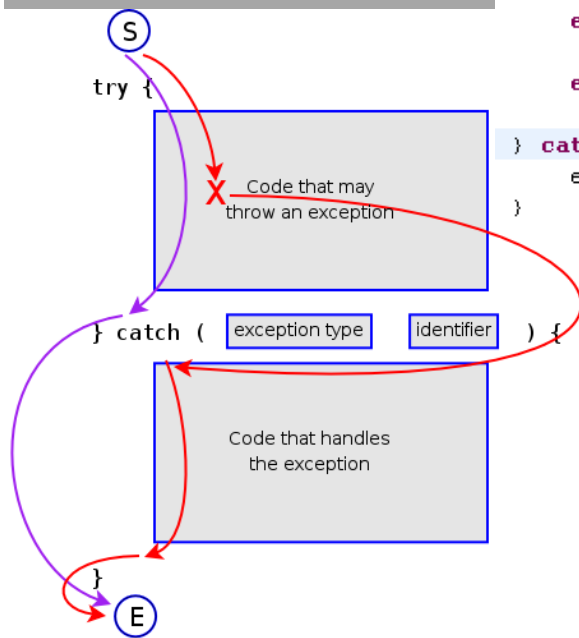
No exceptions thrown:



An exception arises :

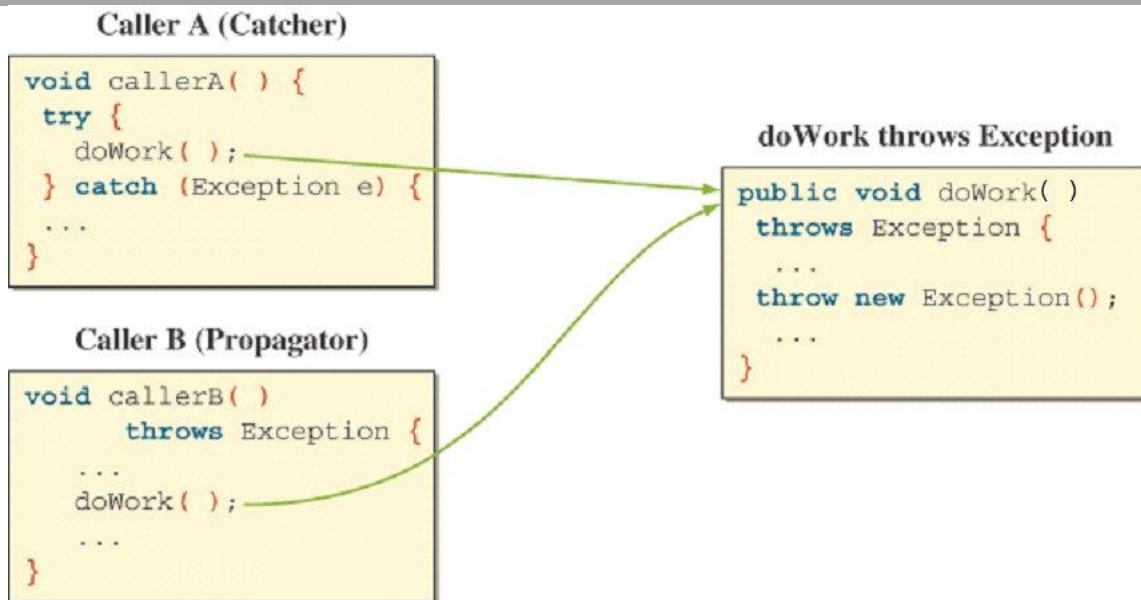


Throw

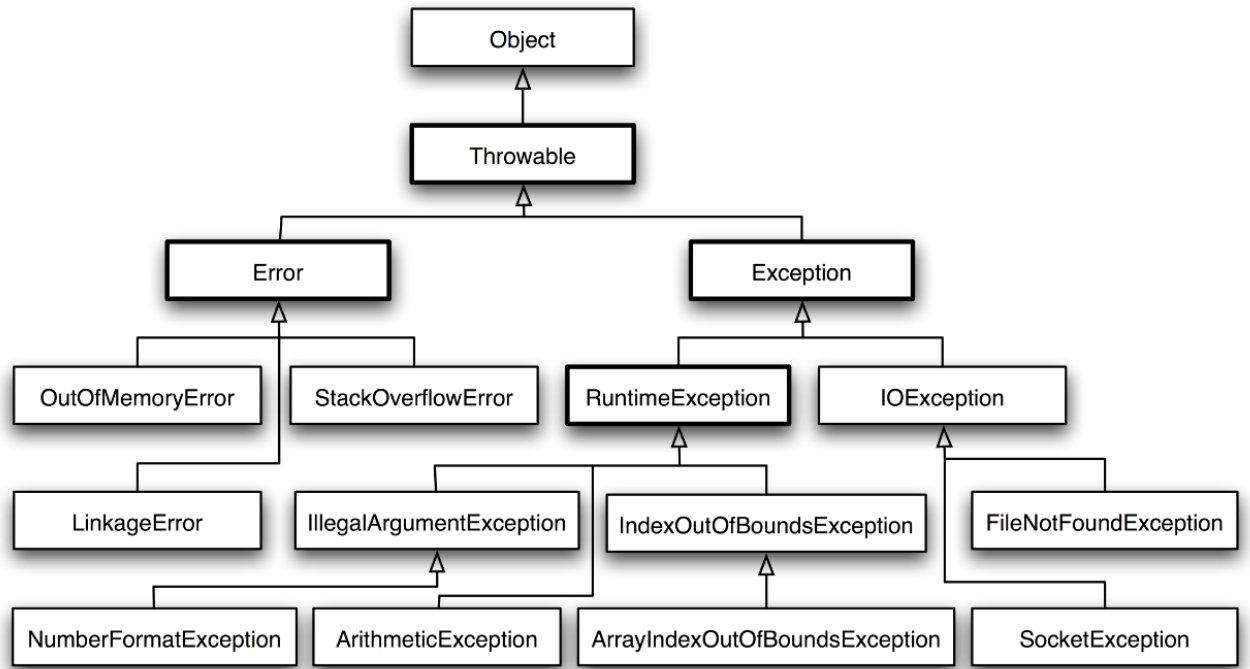


```
try {  
    if (s == null)  
        throw new NullPointerException();  
    else if (s.length() == 0)  
        throw new IllegalArgumentException();  
    else  
        throw new Exception();  
} catch (NullPointerException | IllegalArgumentException e) {  
    e.printStackTrace();  
}
```

Throws



Built in Exception



Custom exception

```
// AgeException.java → 6
package pack; → 1

public class AgeException extends Exception
{
    ↓ 2           ↓ 3
    public AgeException(String s) → 4
    {
        super(s); → 5
    }
}
```

```
// save by CheckAge.java
package nage;

public class CheckAge
{
    public void verify(int age) throws AgeException
    {
        if (age > 0)
        {
            System.err.print("valid age");
        }
        else
        {
            AgeException ae = new AgeException("Invalid age");
            throw ae;
        }
    }
}
```

Throwable Class

Create your own, custom, Throwable...

```
class MyThrowable extends Throwable {  
}  
  
class CustomThrower {  
    public void anotherMethod() throws MyThrowable {  
        throw new MyThrowable();  
    }  
}
```

... that can be used like this.

throw vs throws

throw	throws
1. Java throw keyword is used to explicitly throw an exception	1. Java throws keyword is used to declare an exception.
2. void m(){ throw new ArithmeticException("sorry"); }	2. void m()throws ArithmeticException{ //method code }
3. Checked exception cannot be propagated using throw only.	3. Checked exception can be propagated with throws.
4. Throw is followed by an instance.	4. Throw is followed by a class.
5. Throw is used within the method.	5. Throws is used with the method signature.
6. You cannot throw multiple exceptions.	6. You can declare multiple exceptions e.g. public void method()throws IOException,SQLException.

References:

- <http://journals.ecs.soton.ac.uk/java/tutorial/java/exceptions/throwable.html>
- <http://javaconceptoftheday.com/difference-between-throw-throws-and-throwable-in-java/>
- <http://www.javaworld.com/article/2076700/core-java/exceptions-in-java.html>

Questions/Comments



