Topics

- Writing Your Own Exceptions
- Use of throw and throws Clauses

Writing Your Own Exceptions

- **Programmer Can Write Either a Checked Exception OR an Unchecked Type Exception.**
- **To Create a Checked Type Exception → Make Your Exception class a direct subclass of Exception OR any one of its subclass Except RunTimeException.**

class AException extends Exception { …} → Checked Exception

class BException extends IOException { ..} → Checked Exception

- **To Create an Unchecked Exception → Make Your Exception class a subclass of RuntimeException OR any one of its subclass .**

class XException extends RuntimeException { … }

class YException extends AritmeticException { … }

throw Clause [statement]

- **'throw' clause in Java is used to throw Exceptions**
- **The clause can only be used for Exception classes**
- **Syntax**

1. ***throw ThrowableInstance → Where ThrowableInstance must belong to an Object of Type Throwable or any of its sub class***
2. ***throw new Exception-Name() → Where Exception-Name can be either a Exception or any of its sub-class***
3. ***throw new Exception-Name(parameters) → In this form parameters can be supplied with exception [Assumption: The desired exception class must supplies a parameterized constructor]***

‘throws’ clause in Java

- If a method causes an Exception that it can not handle then it must specify this behavior using throws clause
- Specifically required if a method throws a Checked Type Exception
- Optional → if a method throws an Un-Checked Type Exception
- Syntax

```
return-type  method-name(parameters)     throws     exception-list
{
        ….. Method Body
}
```

Creating Your Own Exceptions [Example : Un-Checked Type]

Define an Un-checked type exception class named 'InvalidBoxDimensionException' which can be thrown whenever an attempt is made to create an instance of class 'Box' with length, width or height is either less than or equal to 0.

```
class InvalidBoxDimensionException extends RuntimeException
{
        InvalidBoxDimensionException (double invalidDim)
        {
                System.out.println("Box Instance with Invalid Dimension :"+ invalidDim);
        }
}// End of class
```

Creating Your Own Exceptions [Example : Un-Checked Type ….]


```
class       Box
{
        private    double    length;
        private    double    width;
        private    double    height;
        Box (double length, double width, double height) throws InvalidBoxDimensionException
        {
                if( length <=0)       throw new InvalidBoxDimensionException(length);
                if( width <=0)        throw new InvalidBoxDimensionException(width);
                if( height<=0)        throw new InvalidBoxDimensionException(height);
                this.length = length; this.width = width; this.height = height;
        }// End of Constructor
        public double area() { return 2*(length*width + width*height + height*length);}
         public double volume() { return length*width*height; }
}// End of class
```

Throws Clause is Optional for Unchecked Exceptions

Creating Your Own Exceptions [Example : Un-Checked Type ….]


```
class       Driver
{
        public    static    void      main(String args[])
        {
                try
                {
                        Box b1 = new Box(10, 0, 56);
                        System.out.println(b1.area();
                }
                catch(InvalidBoxDimensionException e) {}
                try
                {
                        Box b2 = new Box(10, 0, 56);
                        System.out.println(b2.area();
                }
                catch(InvalidBoxDimensionException e) {}
        } // End of Method
}// End of class
```

Creating Your Own Exceptions [Example : Checked Type]

Define a checked type exception class named 'InvalidBoxDimensionException' which can be thrown whenever an attempt is made to create an instance of class 'Box' with length, width or height is either less than or equal to 0.

```
class InvalidBoxDimensionException extends Exception
{
        InvalidBoxDimensionException (double invalidDim)
        {
                System.out.println("Box Instance with Invalid Dimension :"+ invalidDim);
        }
}// End of class
```

Creating Your Own Exceptions [Example : Checked Type ….]


```
class       Box
{
        private    double    length;
        private    double    width;
        private    double    height;
        Box (double length, double width, double height) throws InvalidBoxDimensionException
        {
                if( length <=0)       throw new InvalidBoxDimensionException(length);
                if( width <=0)        throw new InvalidBoxDimensionException(width);
                if( height<=0)        throw new InvalidBoxDimensionException(height);
                this.length = length; this.width = width; this.height = height;
        }// End of Constructor
        public double area() { return 2*(length*width + width*height + height*length);}
         public double volume() { return length*width*height; }
}// End of class
```

Throws Clause is Must for Checked Exceptions

Creating Your Own Exceptions [Example : Checked Type ….]


```
class       Driver
{
        public    static    void      main(String args[])
        {
                try
                {
                        Box b1 = new Box(10, 0, 56);
                        System.out.println(b1.area();
                }
                catch(InvalidBoxDimensionException e) {}
                try
                {
                        Box b2 = new Box(10, 0, 56);
                        System.out.println(b2.area();
                }
                catch(InvalidBoxDimensionException e) {}
        } // End of Method
}// End of class
```

Thank You