

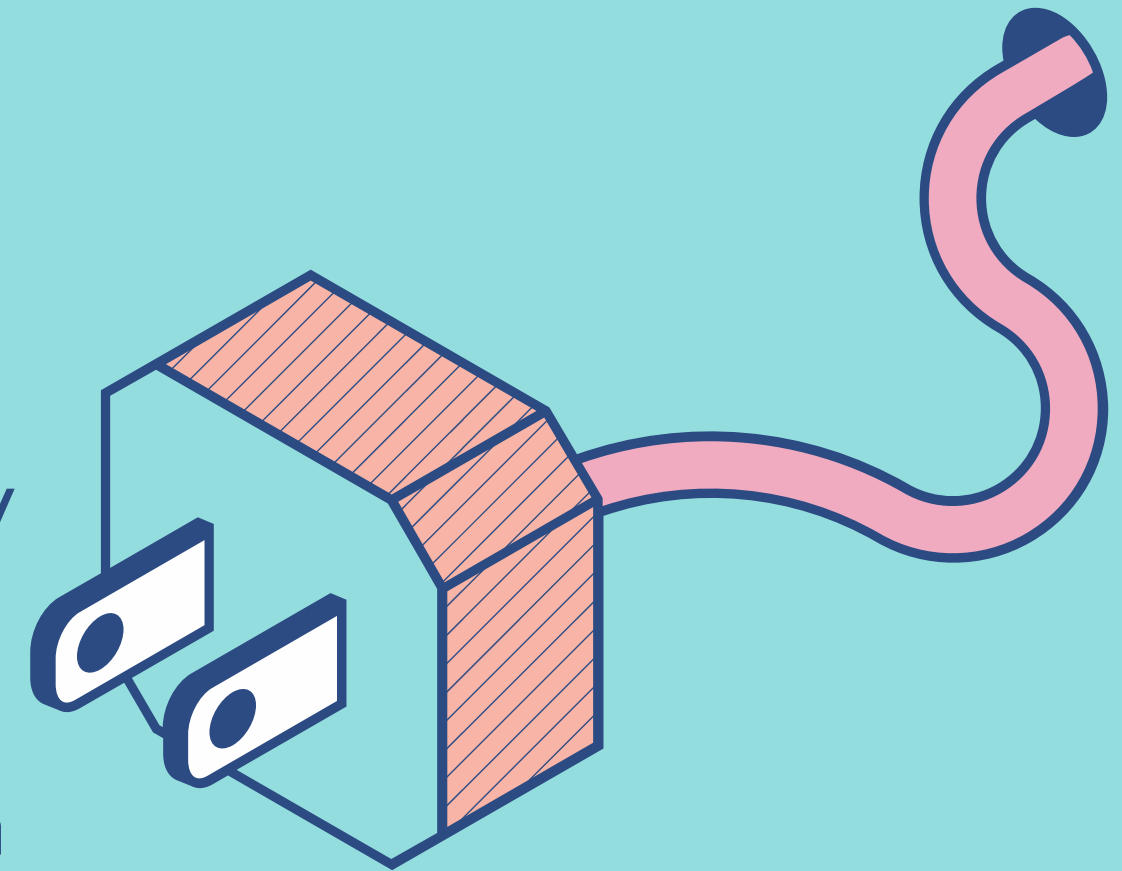


Exception Classes

All exception types are objects of classes derived from the "Throwable" class defined in the java.lang package.

Throwable

The Throwable class is the superclass of all errors and exceptions in the Java language. Only objects that are instances of this class (or one of its subclasses) are thrown by the Java virtual machine or can be thrown by the Java throw statement. Similarly, only this class or one of its subclasses can be the argument type in a catch clause. For the purposes of compile-time exception checking, Throwable and any subclass of Throwable that is not also a subclass of either RuntimeException or Error are considered checked exceptions.





Exception

This class is used for exceptional conditions that program users must catch.

It covers all exceptions that a program can handle. Is the class from which subclasses are derived needed to create your own exception types.

RuntimeException

Are required exceptions because they don't have to be handled compulsorily. This class covers exceptions that occur when executing on the data that manipulates the application and that reside in memory, for example:

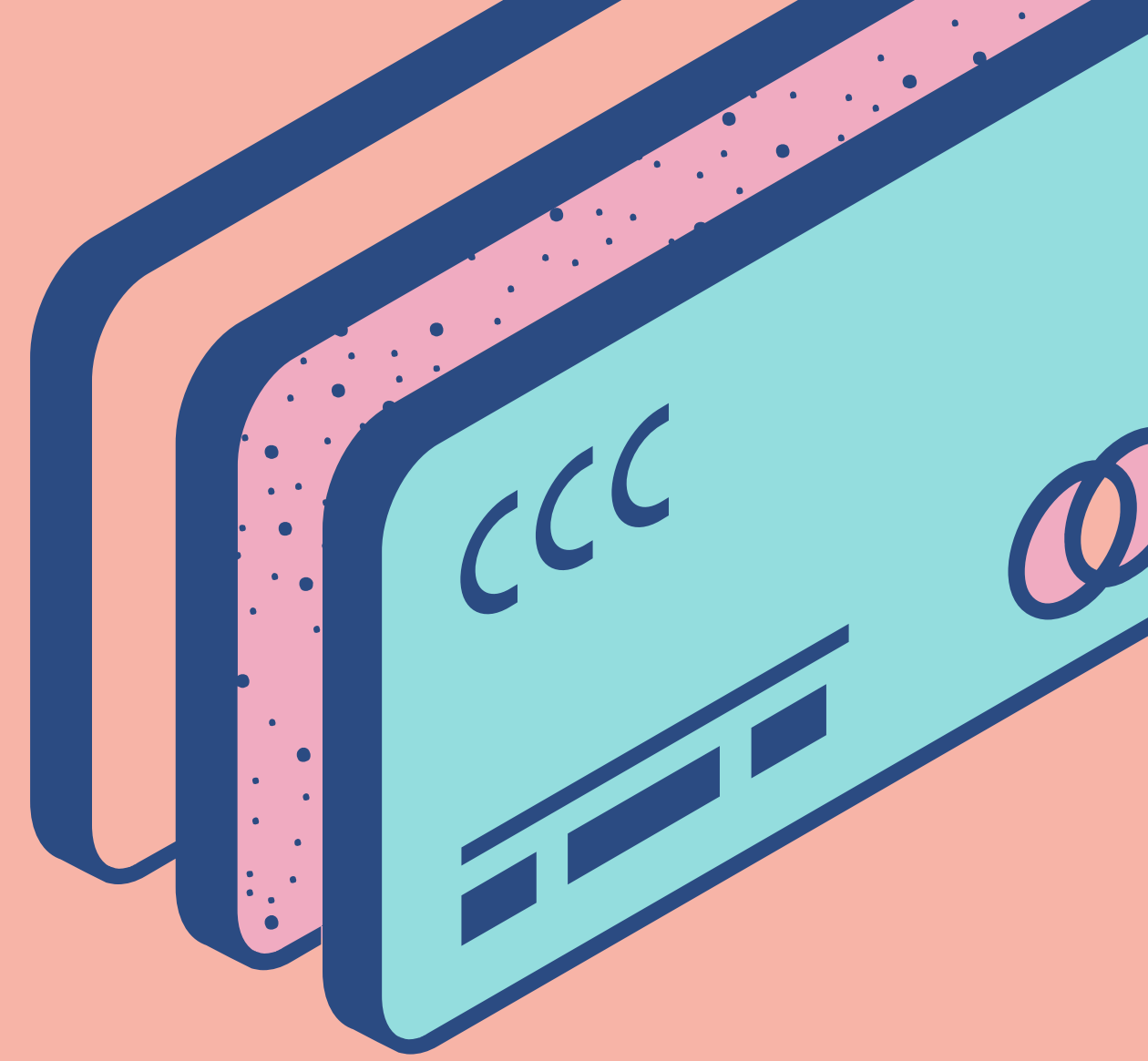
- **ArithmeticException**
- **NullPointerException**
- **IndexOutOfBoundsException**
- **NumberFormatException**

1.-ArithmeticException

IS THE BASE CLASS FOR THE FOLLOWING
EXCEPTIONS:

DivideByZeroException

Which is thrown in integer division when the divisor is 0. For example, if you try to divide 10 by 0, a DivideByZeroException is thrown.



```
1 package Main;
2
3 public class Main {
4
5     public static int numerador = 10;
6     public static Integer denominador = 0;
7     public static float division;
8
9     public static void main(String[] args) {
10         System.out.println("ANTES DE HACER LA DIVISIÓN");
11         try {
12             division = numerador / denominador;
13         } catch (ArithmeticException ex) {
14             division = 0; // Si hay una excepción doy valor '0' al atributo 'division'
15             System.out.println("Error: "+ex.getMessage());
16         } finally {
17             System.out.println("División: "+division);
18             System.out.println("DESPUES DE HACER LA DIVISIÓN");
19         }
20     }
21 }
```

Problems @ Javadoc Declaration Console

<terminated> Main (5) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_20.jdk/Contents/Home/bin/java (1

ANTES DE HACER LA DIVISIÓN
Error: / by zero
División: 0.0
DESPUES DE HACER LA DIVISIÓN

NotFiniteNumberException

When an arithmetic operation produces a result that is outside the range of allowed data.

```
5 public class Exception {  
6     public static void main(String[] args) {  
7         double x = Double.POSITIVE_INFINITY;  
8         if (!Double.isFinite(x)) {  
9             throw new NotFiniteNumberException(value: x);  
10        }  
11    }  
12 }
```

Output - Exception (run) x

```
run:  
Exception in thread "main" exception.NotFiniteNumberException: Value is not finite: Infinity  
    at exception.Exception.main(Exception.java:9)  
C:\Users\User\AppData\Local\NetBeans\Cache\17\executor-snippets\run.xml:111: The following error occurred while executing this  
C:\Users\User\AppData\Local\NetBeans\Cache\17\executor-snippets\run.xml:68: Java returned: 1  
BUILD FAILED (total time: 0 seconds)
```

OverflowException

When an arithmetic operation produces a result that is outside the range of allowed data.

```
4
5 public class EjemploOverflowException {
6     public static void main(String[] args) {
7         int x = Integer.MAX_VALUE;
8         int y = x + 1;
9         if (y < x) {
10             throw new OverflowException(message: "Integer overflow occurred");
11         }
12     }
13 }
14
15
```

Output - Exception (run) ×

```
run:
Exception in thread "main" exception.OverflowException: Integer overflow occurred
    at exception.EjemploOverflowException.main(EjemploOverflowException.java:10)
C:\Users\User\AppData\Local\NetBeans\Cache\17\executor-snippets\run.xml:111: The following error occurred while executing this
C:\Users\User\AppData\Local\NetBeans\Cache\17\executor-snippets\run.xml:68: Java returned: 1
BUILD FAILED (total time: 0 seconds)
```


2.- NullPointerException

Is an exception that is thrown when an application tries to use null when an object is required. And this includes calling an instance method of a null object.

```
public class EjemploNullPointerException {  
    public static void main(String[] args) {  
        String str = null;  
        int length = str.length();  
    }  
}
```

Output - Exception (run) X

run:

```
Exception in thread "main" java.lang.NullPointerException: Cannot invoke "String.length()" because "str" is null  
    at exception.EjemploNullPointerException.main(EjemploNullPointerException.java:7)
```

```
C:\Users\User\AppData\Local\NetBeans\Cache\17\executor-snippets\run.xml:111: The following error occurred while executing this task: java.lang.NullPointerException: Cannot invoke "String.length()" because "str" is null
```

```
C:\Users\User\AppData\Local\NetBeans\Cache\17\executor-snippets\run.xml:94: Java returned: 1
```

```
BUILD FAILED (total time: 1 second)
```

3.-IndexOutOfBoundsException

The exception that is thrown when you try to access the wrong index of the array. That is, a negative index or an index above the size of the array.

The screenshot shows an IDE with a Java source file and an output window. The source file contains the following code:

```
1 package exception;
2
3
4 public class EjemploIndexOutOfBoundsException {
5     public static void main(String[] args) {
6         int[] numbers = {1, 2, 3};
7         int x = numbers[3];
8     }
9 }
10
```

The output window, titled "Output - Exception (run)", shows the following error message:

```
run:
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3
    at exception.EjemploIndexOutOfBoundsException.main(EjemploIndexOutOfBoundsException.java:7)
C:\Users\User\AppData\Local\NetBeans\Cache\17\executor-snippets\run.xml:111: The following error occurred while executing this
C:\Users\User\AppData\Local\NetBeans\Cache\17\executor-snippets\run.xml:68: Java returned: 1
BUILD FAILED (total time: 0 seconds)
```

4.- NumberFormatException

This exception is thrown when trying to convert a malformed string to a numeric value. That means, when it is not possible to convert a string to any numeric type (float, int, etc.), this exception is thrown.

```
public class EjemploNumberFormatException {  
    public static void main(String[] args) {  
        String str = "not a number";  
        int num = Integer.parseInt(str);  
    }  
}
```

out - Exception (run) X

run:

```
Exception in thread "main" java.lang.NumberFormatException: For input string: "not a number"  
    at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:67)  
    at java.base/java.lang.Integer.parseInt(Integer.java:665)  
    at java.base/java.lang.Integer.parseInt(Integer.java:781)  
    at exception.EjemploNumberFormatException.main(EjemploNumberFormatException.java:7)
```

[C:\Users\User\AppData\Local\NetBeans\Cache\17\executor-snippets\run.xml:111](#): The following error occurred while executing this task:

[C:\Users\User\AppData\Local\NetBeans\Cache\17\executor-snippets\run.xml:94](#): Java returned: 1

BUILD FAILED (total time: 1 second)