Exception handling: capture

1. Exception handling: Introduction
2. Exception capture

Exception handling: Introduction

Exception handling keywords

try

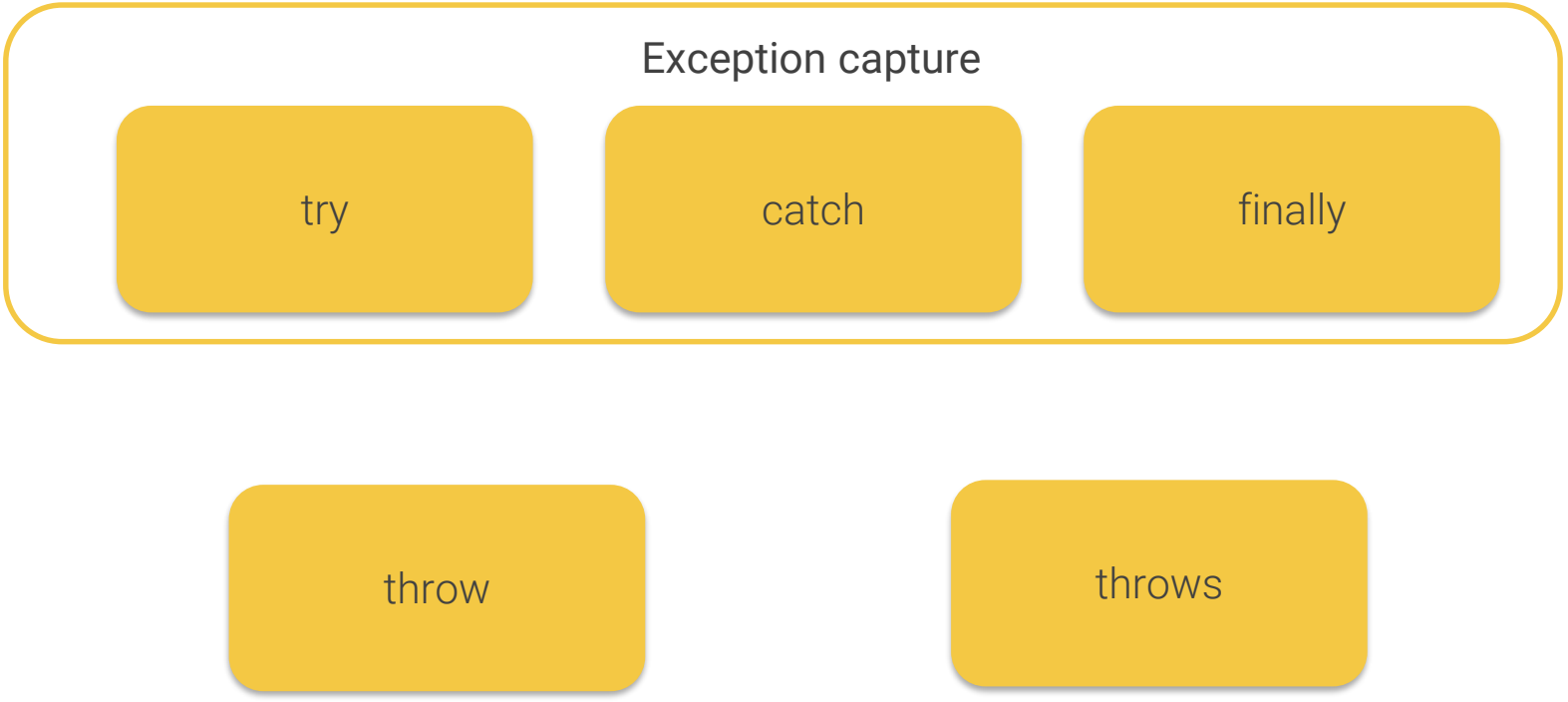
catch

finally

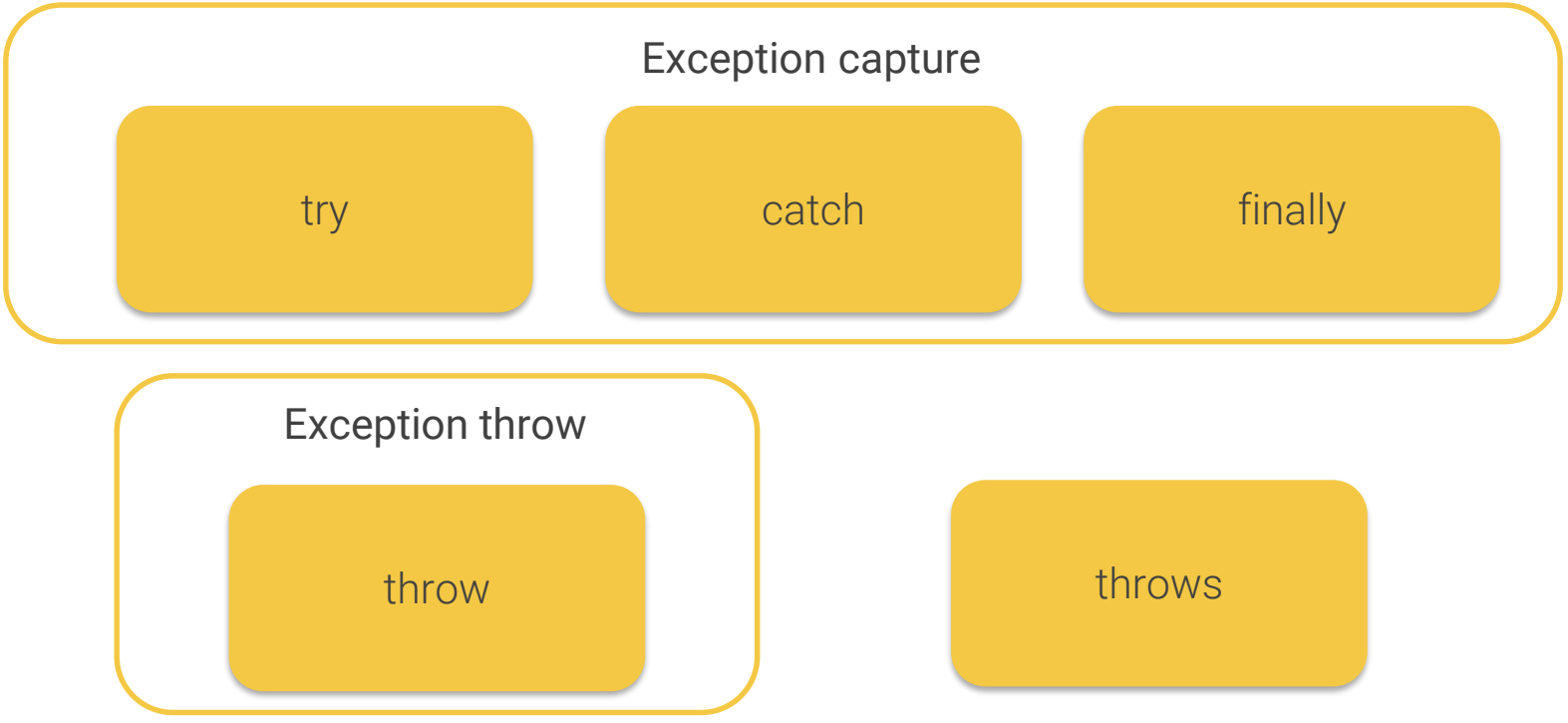
throw

throws

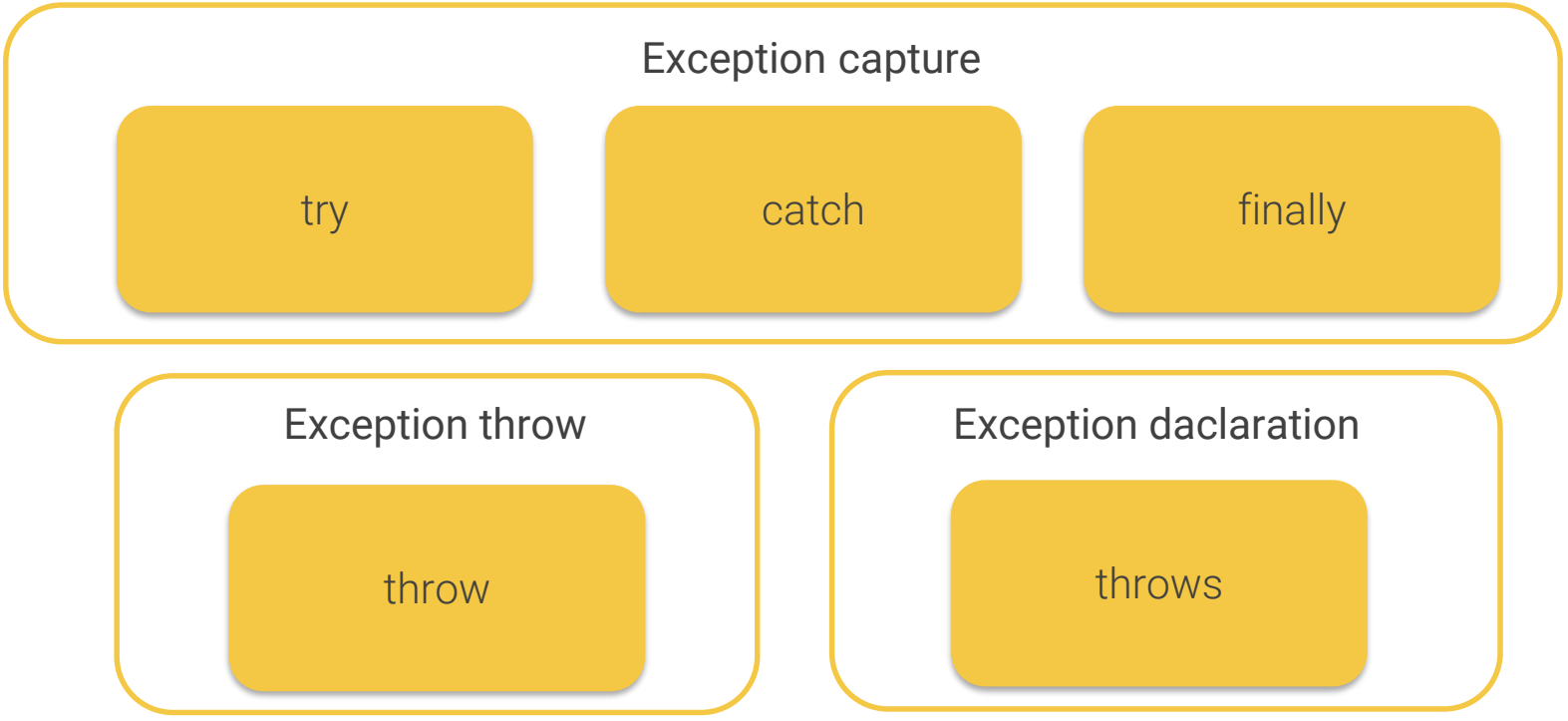
Exception handling keywords



Exception handling keywords



Exception handling keywords



Exception capture

try, catch & finally clauses

```
try{
     //code that may cause an Exception
}catch (nombreClasseExcepcion1 e1){
     //actions to do when de Exception1 happens
}catch (nombreClasseExcepcion2 e2){
     //actions to do when de Exception2 happens
} finally{
     //code to be executed always
}
```

try, catch & finally clauses

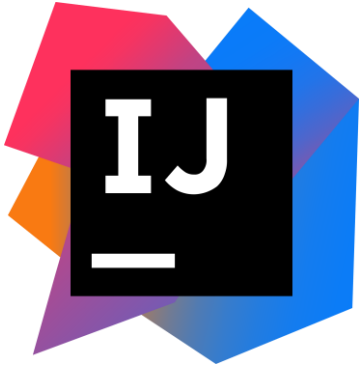
```
try{
     //code that may cause an Exception
}catch (nombreClasseExcepcion1 e1){
     //Actions to do when de Exception1 happens
}catch (nombreClasseExcepcion2 e2){
     //Actions to do when de Exception2 happens
} finally{
     //code to be executed always
}
```

try, catch & finally clauses

```
try{
     //code that may cause an Exception
}catch (nombreClasseExcepcion1 e1){
     //Actions to do when de Exception1 happens
}catch (nombreClasseExcepcion1 e2){
     //Actions to do when de Exception2 happens
}finally{
     //code to be executed always
}
```

try, catch & finally clauses

```
try{
     //code that may cause an Exception
}catch (nombreClasseExcepcion1 e1){
     //Actions to do when de Exception1 happens
}catch (nombreClasseExcepcion2 e2){
     //Actions to do when de Exception2 happens
}finally{
     //code to be executed always
}
```



TRY-CATCH-FINALLY DEMO

2. Exception capture

```
public static void main(String[] args) {
       InputStream in = null;
       try {
           System.out.println("About to open a file");
           in = new FileInputStream("missingfile.txt");
           System.out.println("File open");
           int s = in.read();
       } catch(FileNotFoundException e){
           System.out.println(e.getClass().getName()+"-"+ e.getMessage());
       } catch(IOException e){
           System.out.println(e.getClass().getName()+"-"+ e.getMessage());
       } finally{
           System.out.println("Application end");
           try{
               if(in != null){
                   in.close();
               }
           }catch(IOException e){
               System.out.println("Failed to close file");
           }
       }
   }
```

2. Exception capture

```
public static void main(String[] args) {
        InputStream in = null;
        try {
            System.out.println("About to open a file");
            in = new FileInputStream("missingfile.txt");
            System.out.println("File open");
            int s = in.read();
        } catch(FileNotFoundException e){
            System.out.println(e.getClass().getName()+"-"+ e.getMessage());
        } catch(IOException e){
            System.out.println(e.getClass().getName()+"-"+ e.getMessage());
        } finally{
            System.out.println("Application end");
            try{
                if(in != null){
                    in.close();
                }
            }catch(IOException e){
                System.out.println("Failed to close file");
            }
        }
    }
```

2. Exception capture

```
public static void main(String[] args) {
        InputStream in = null;
        try {
            System.out.println("About to open a file");
            in = new FileInputStream("missingfile.txt");
            System.out.println("File open");
            int s = in.read();
        } catch(FileNotFoundException e){
            System.out.println(e.getClass().getName()+"-"+ e.getMessage());
        } catch(IOException e){
            System.out.println(e.getClass().getName()+"-"+ e.getMessage());
        } finally{
            System.out.println("Application end");
            try{
                if(in != null){
                    in.close();
                }
            }catch(IOException e){
                System.out.println("Failed to close file");
            }
        }
    }
```

java.io.FileNotFoundException

java.io.IOException

2. Exception capture

```
public static void main(String[] args) {
        InputStream in = null;
        try {
            System.out.println("About to open a file");
            in = new FileInputStream("missingfile.txt");
            System.out.println("File open");
            int s = in.read();
        } catch(FileNotFoundException e){
            System.out.println(e.getClass().getName()+"-"+ e.getMessage());
        } catch(IOException e){
            System.out.println(e.getClass().getName()+"-"+ e.getMessage());
        } finally{
            System.out.println("Application end");
            try{
                if(in != null){
                    in.close();
                }
            }catch(IOException e){
                System.out.println("Failed to close file");
            }
        }
    }
```

java.io

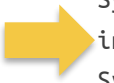
Class FileNotFoundException

java.lang.Object
- java.lang.Throwable
  - java.lang.Exception
    - java.io.IOException
      - java.io.FileNotFoundException

2. Exception capture

```
public static void main(String[] args) {
        InputStream in = null;
        try {
            System.out.println("About to open a file");
            in = new FileInputStream("missingfile.txt");
            System.out.println("File open");
            int s = in.read();
        } catch(FileNotFoundException e){
            System.out.println(e.getClass().getName()+"-"+ e.getMessage());
        } catch(IOException e){
            System.out.println(e.getClass().getName()+"-"+ e.getMessage());
        } finally{
            System.out.println("Application end");
            try{
                if(in != null){
                    in.close();
                }
            }catch(IOException e){
                System.out.println("Failed to close file");
            }
        }
    }
```

2. Exception capture



```
public static void main(String[] args) {
        InputStream in = null;
        try {
            System.out.println("About to open a file");
            in = new FileInputStream("missingfile.txt");
            System.out.println("File open");
            int s = in.read();
        } catch(FileNotFoundException e){
            System.out.println(e.getClass().getName()+"-"+ e.getMessage());
        } catch(IOException e){
            System.out.println(e.getClass().getName()+"-"+ e.getMessage());
        } finally{
            System.out.println("Application end");
            try{
                if(in != null){
                    in.close();
                }
            }catch(IOException e){
                System.out.println("Failed to close file");
            }
        }
    }
```

2. Exception capture

```
public static void main(String[] args) {
        InputStream in = null;
        try {
            System.out.println("About to open a file");
            in = new FileInputStream("missingfile.txt");
            System.out.println("File open");
            int s = in.read();
        } catch(FileNotFoundException e){
            System.out.println(e.getClass().getName()+"-"+ e.getMessage());
        } catch(IOException e){
            System.out.println(e.getClass().getName()+"-"+ e.getMessage());
        } finally{
            System.out.println("Application end");
            try{
                if(in != null){
                    in.close();
                }
            }catch(IOException e){
                System.out.println("Failed to close file");
            }
        }
    }
```

java.io.IOException

2. Exception capture

```
public static void main(String[] args) {
        InputStream in = null;
        try {
            System.out.println("About to open a file");
            in = new FileInputStream("missingfile.txt");
            System.out.println("File open");
            int s = in.read();
        } catch(FileNotFoundException e){
            System.out.println(e.getClass().getName()+"-"+ e.getMessage());
        } catch(IOException e){
            System.out.println(e.getClass().getName()+"-"+ e.getMessage());
        } finally{
            System.out.println("Application end");
            try{
                if(in != null){
                    in.close();
                }
            }catch(IOException e){
                System.out.println("Failed to close file");
            }
        }
    }
```

java.io.IOException

multi-catch

```
try {
     System.out.println("About to open a file");
     in = new FileInputStream("missingfile.txt");
     System.out.println("File open");
     int x = in.read();
     int y = 20 / x
} catch(ArithmeticException e){
     System.out.println(e.getClass().getName()+"-"+ e.getMessage());
} catch(FileNotFoundException e){
     System.out.println(e.getClass().getName()+"-"+ e.getMessage());
} catch(IOException e){
     System.out.println(e.getClass().getName()+"-"+ e.getMessage());
}
```

java.lang.ArithmeticException

multi-catch

```
try {
     System.out.println("About to open a file");
     in = new FileInputStream("missingfile.txt");
     System.out.println("File open");
     int x = in.read();
     int y = 20 / x
} catch(FileNotFoundException e){
     System.out.println(e.getClass().getName()+"-"+ e.getMessage());
} catch(IOException e){
     System.out.println(e.getClass().getName()+"-"+ e.getMessage());
} catch(ArithmeticException e){
     System.out.println(e.getClass().getName()+"-"+ e.getMessage());
}
```

java.lang.ArithmeticException

multi-catch

```
try {
     System.out.println("About to open a file");
     in = new FileInputStream("missingfile.txt");
     System.out.println("File open");
     int x = in.read();
     int y = 20 / x
} catch(FileNotFoundException e){
     System.out.println(e.getClass().getName()+"-"+ e.getMessage());
} catch(IOException e){
     System.out.println(e.getClass().getName()+"-"+ e.getMessage());
} catch(ArithmeticException e){
     System.out.println(e.getClass().getName()+"-"+ e.getMessage());
}
```

multi-catch

```
try {
     System.out.println("About to open a file");
     in = new FileInputStream("missingfile.txt");
     System.out.println("File open");
     int x = in.read();
     int y = 20 / x
} catch(FileNotFoundException | ArithmeticException e){
     System.out.println(e.getClass().getName()+"-"+ e.getMessage());
} catch(IOException e){
     System.out.println(e.getClass().getName()+"-"+ e.getMessage());
}
```

multi-catch

```
try {
     System.out.println("About to open a file");
     in = new FileInputStream("missingfile.txt");
     System.out.println("File open");
     int x = in.read();
     int y = 20 / x
} catch(FileNotFoundException | ArithmeticException e){
     System.out.println(e.getClass().getName()+"-"+ e.getMessage());
} catch(IOException e){
     System.out.println(e.getClass().getName()+"-"+ e.getMessage());
}
```

multi-catch

```
try {
     System.out.println("About to open a file");
     in = new FileInputStream("missingfile.txt");
     System.out.println("File open");
     int x = in.read();
     int y = 20 / x
} catch(FileNotFoundException | IOException | ArithmeticException e){
     System.out.println(e.getClass().getName()+"-"+ e.getMessage());
}
```

Types in multi-cast must be disjoint: FileNotFoundException is a subclass of IOException

“A excepción del hombre, ningún ser se maravilla de su propia existencia.”

Arthut Schopenhauer Filósofo alemán