

JAVA

Pulse para añadir texto

Arreglos y Excepciones



TOGETHER. FREE YOUR ENERGIES

Objetivo

- Lograr que aprendamos a programar en Java, hasta el punto de construir aplicaciones reales en la tecnología.



Agenda de hoy

- ✓ Parte 1: Herramientas de base y J2EE

Módulo 1:

- ✓ El lenguaje Java
 - ✓ Arreglos
 - ✓ Excepciones

Arrays I

Se declaran de la siguiente forma:

```
tipoDeDato[] variable = new tipoDeDato[tamaño];  
tipoDeDato[] variable = {valor1, valor2, .... , valorN}
```

```
String[] arreglo = new String[10]; //un arreglo de 10 strings
```

Para acceder a un elemento del Array, utilizamos:

```
variable[posicion]
```

```
String elPrimero = arreglo[0];
```

Si intentamos acceder a una posición fuera del array, se produce una excepción

Arrays II

Para conocer la longitud de un Array, podemos acceder a su atributo público **length**.

El índice de un Array es de tipo int.

Arrays III

Ejemplo:

```
public class Colecciones
{
    public static void main(String[] args)
    {
        // Creación e inicialización.
        String[] saludos = new String[4];

        // Inserción.
        saludos[0] = new String("Hola");
        saludos[1] = new String("Adios");
        saludos[2] = new String("Hello");
        saludos[3] = new String("GoodBye");

        // Extracción.
        String tmp = saludos[2];

        // Borrado.
        saludos[2] = null;
        saludos[3] = null;

        // Recorrido.
        for(int i=0; i<saludos.length; i++)
            System.out.println(saludos[i]);

        // Búsqueda.
        boolean sw = false;
        for(int i=0; i<saludos.length; i++)
        {
            if(saludos[i] != null && saludos[i].equals("Adios"));
            {
                System.out.println("Adios ha sido
                    encontrado en la posición: " + i);
                sw = true;
                break;
            }
        }
    }
}
```

Arrays IV

Se pueden crear Arrays en cuyas posiciones se almacenen nuevos Arrays:

Para crear un Array de dos dimensiones:

```
tipoDeDato [ ] [ ] variable = new tipoDeDato [tamaño1] [tamaño2];
```

```
int[][] datos = new int[4][2];
```

También es posible crear arrays bidimensionales pero no cuadrados.

```
int[][] numeros = new int[4][]; //No indicamos la 2ªDimensión  
numeros[0] = new int[2];  
numeros[1] = new int[4];  
numeros[2] = new int[10]; .....
```

Ejercicio



Ejercicio VII de la guía

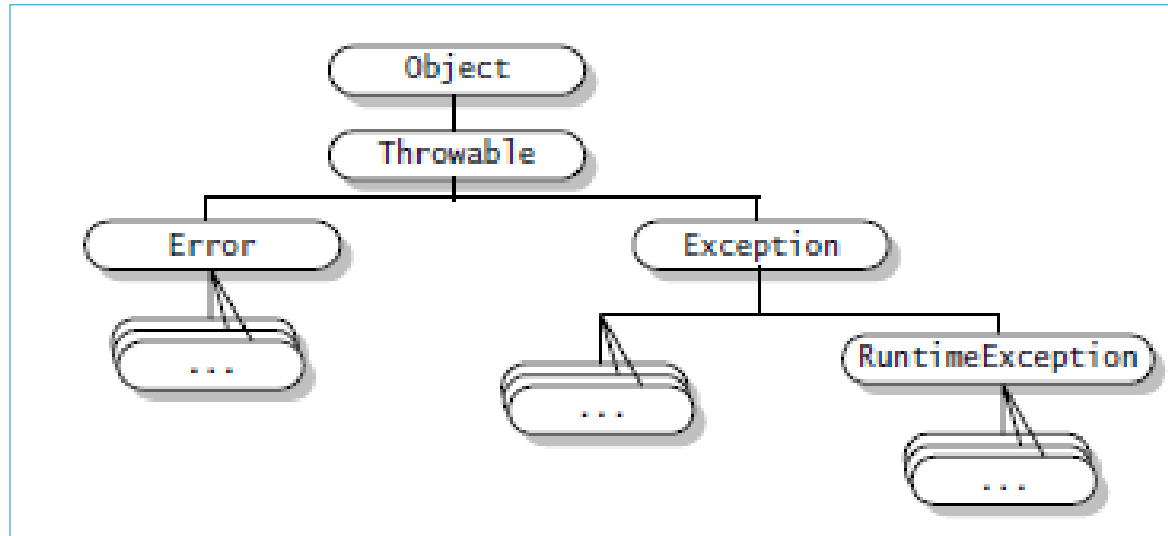


Excepciones I

Las excepciones son la manera que ofrece Java de manejar los errores en tiempo de ejecución.

Las excepciones nos permiten indicar el código que se ejecutará en el caso de producirse un error y continuar con la ejecución del programa, si lo estimamos conveniente.

Excepciones II



Throwable

Superclase que engloba a todas las excepciones.

Error

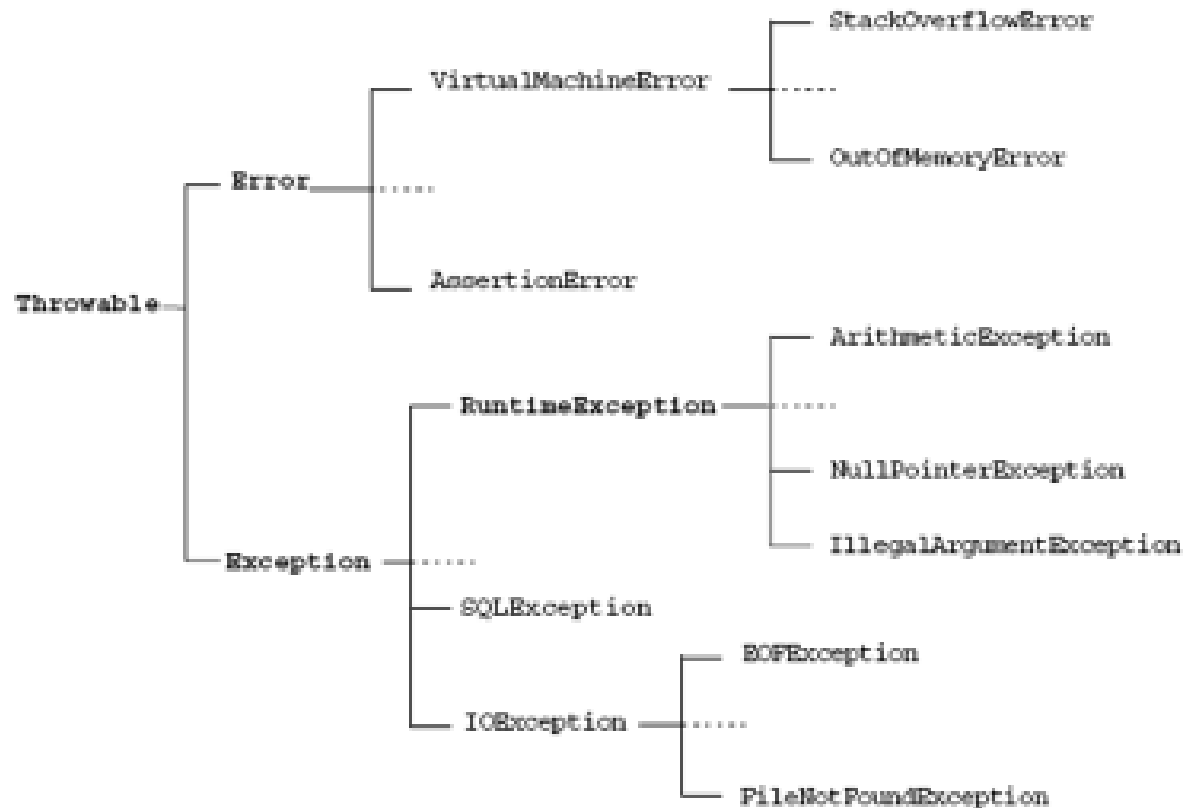
Representa a los errores graves provocados por el sistema: `OutOfMemoryError`, `InternalError`, `UnknownError`,...

Exception

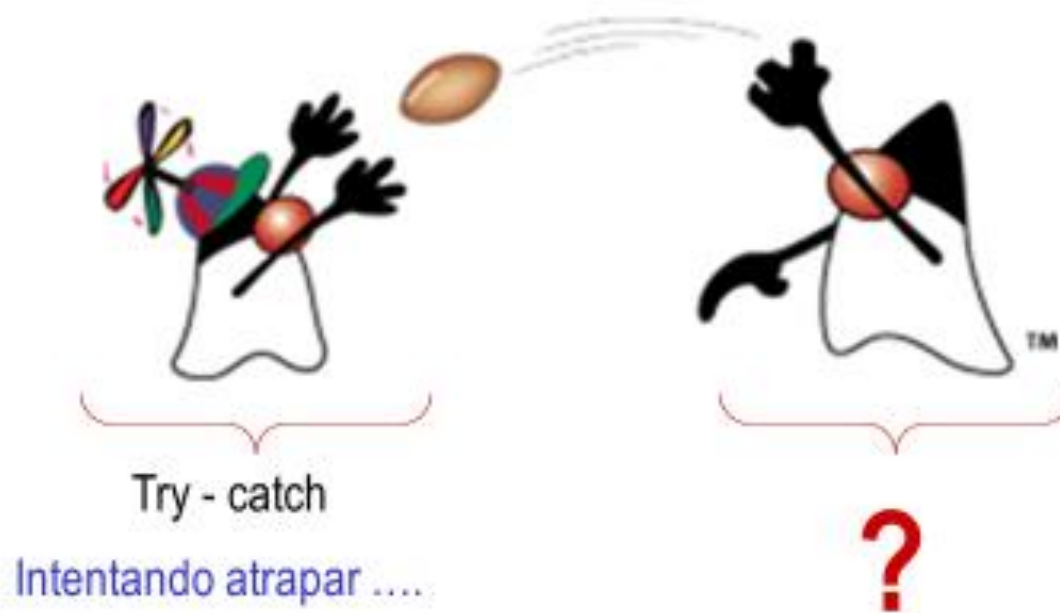
Define las excepciones que los programas deberían controlar obligatoriamente, salvo que hereden de `RuntimeException`



Categorias de Exception



THROWS

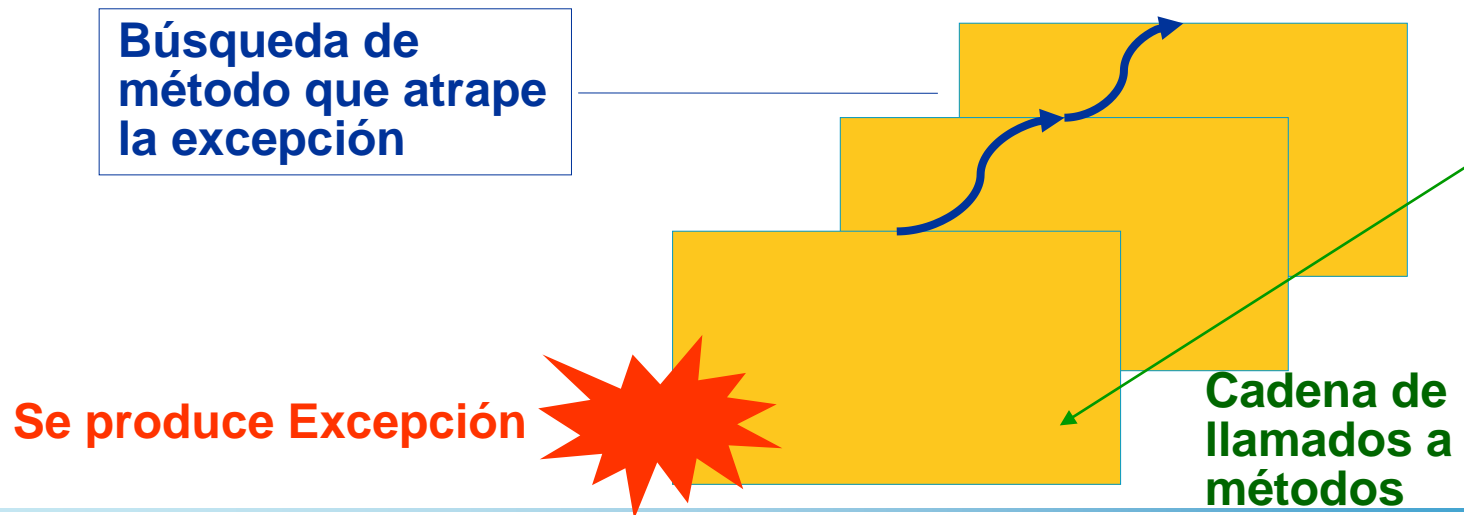


Excepciones III

- En un método, qué se puede hacer si se produce una excepción?

Atrapar la excepción y manejarla

No calentarse, que se preocupe otro (método)



Excepciones IV

Bloque try – catch – finally

Atrapar la excepción y manejarla

Se utiliza para detectar cuándo se produce una excepción y en tal caso, indicar las instrucciones que se deban ejecutar.

Si en la ejecución del código dentro del bloque try se produce una excepción de tipo `TipoExcepcion` (o descendiente de éste),

- se interrumpe la ejecución del resto del código en el bloque try
- se ejecuta el código situado en el bloque catch

```
try {  
    // Código que puede hacer que se eleve la excepción  
}  
catch(TipoExcepcion e) {  
    // Gestor de la excepción  
}
```

Excepciones V

Bloque try – catch – finally

Atrapar la excepción y manejarla

```
public class EjemploCatch {  
    String mensajes[] = {"Primero", "Segundo", "Tercero" };  
    public static void main(String[] args) {  
        try {  
            for(int i = 0; i <= 3; i++)  
                System.out.println(mensajes[i]);  
        }  
        catch ( ArrayIndexOutOfBoundsException e ) {  
            System.out.println("El asunto se nos ha desbordado");  
        }  
        finally {  
            System.out.println("Ha finalizado la ejecución");  
        }  
    }  
}
```

Excepciones VI

Atrapando Excepciones

Atrapar la excepción y manejarla

Se pueden colocar varios bloques catch. Se comprobará si matchea en el mismo orden en que se encuentren esos bloques catch

Sólo se ejecuta un bloque catch. En cuanto se captura la excepción se deja de comprobar el resto de los bloques.

Incorrrrrrrecto

```
catch(Exception e) {  
    ...  
}  
catch(DerivadaDeException e) {  
    ...  
}
```


Excepciones VII

Ejemplo

Atrapar la excepción y manejarla

```
public class Excepcion1 {  
    public static void main(String args[ ]){  
        try {  
            int a = args.length;  
            System.out.println("a = " + a);  
            int b = 42 / a;  
        }  
        catch (ArithmeticException e) {  
            System.out.println("No dividas por 0 (" + e + ")");  
        }  
        System.out.println("La ejecución sigue ...");  
    }  
}
```

Se produce el error

Se captura la excepción

Y sigue la ejecución

Excepciones VIII

No calentarse, que se preocupe otro

Cláusula throws

Salvo que la exception herede de RuntimeException, el compilador exige que si un método no la atrapa DEBE decir explícitamente que no se calienta. Más formalmente, que continúa propagando esa Exception.

```
public String leer(FileInputStream archivo) throws IOException
{
    // ...
}
```

```
public Image cargar(String s) throws EOFException,
                                   MalformedURLException
{
    // ...
}
```

Excepciones IX

Lanzar 'manualmente' excepciones

Las excepciones se lanzan utilizando la sentencia **throw**.

```
public Persona personaEnPosicion(int i) throws IndexOutOfBoundsException{
    Persona p = null;
    if (i<0 || i>this.listaPersonas.size())
        throw new IndexOutOfBoundsException();

    p=this.listaPersonas.get(i);
    return p;
}
```

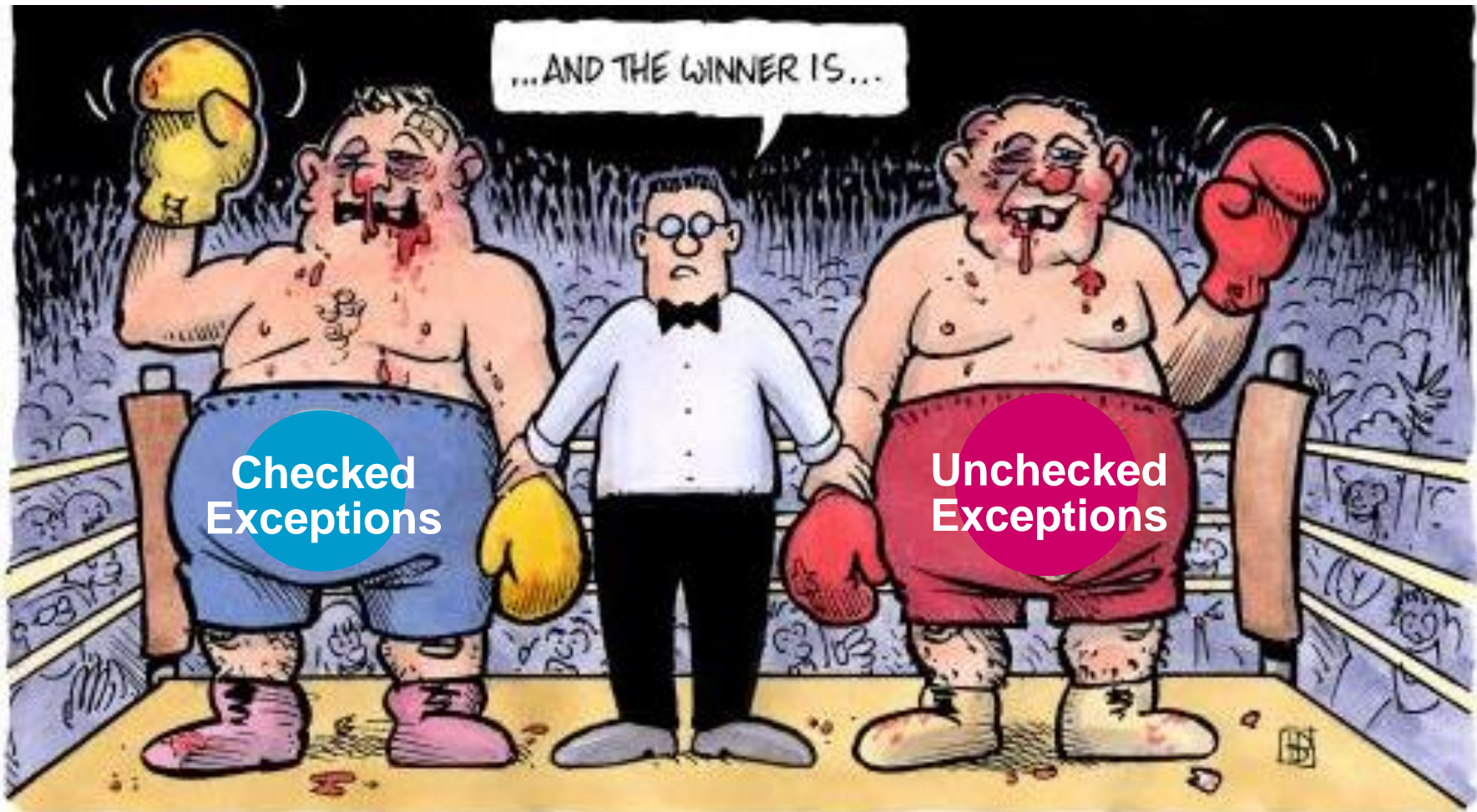
Excepciones X

Excepciones definadas por el usuario

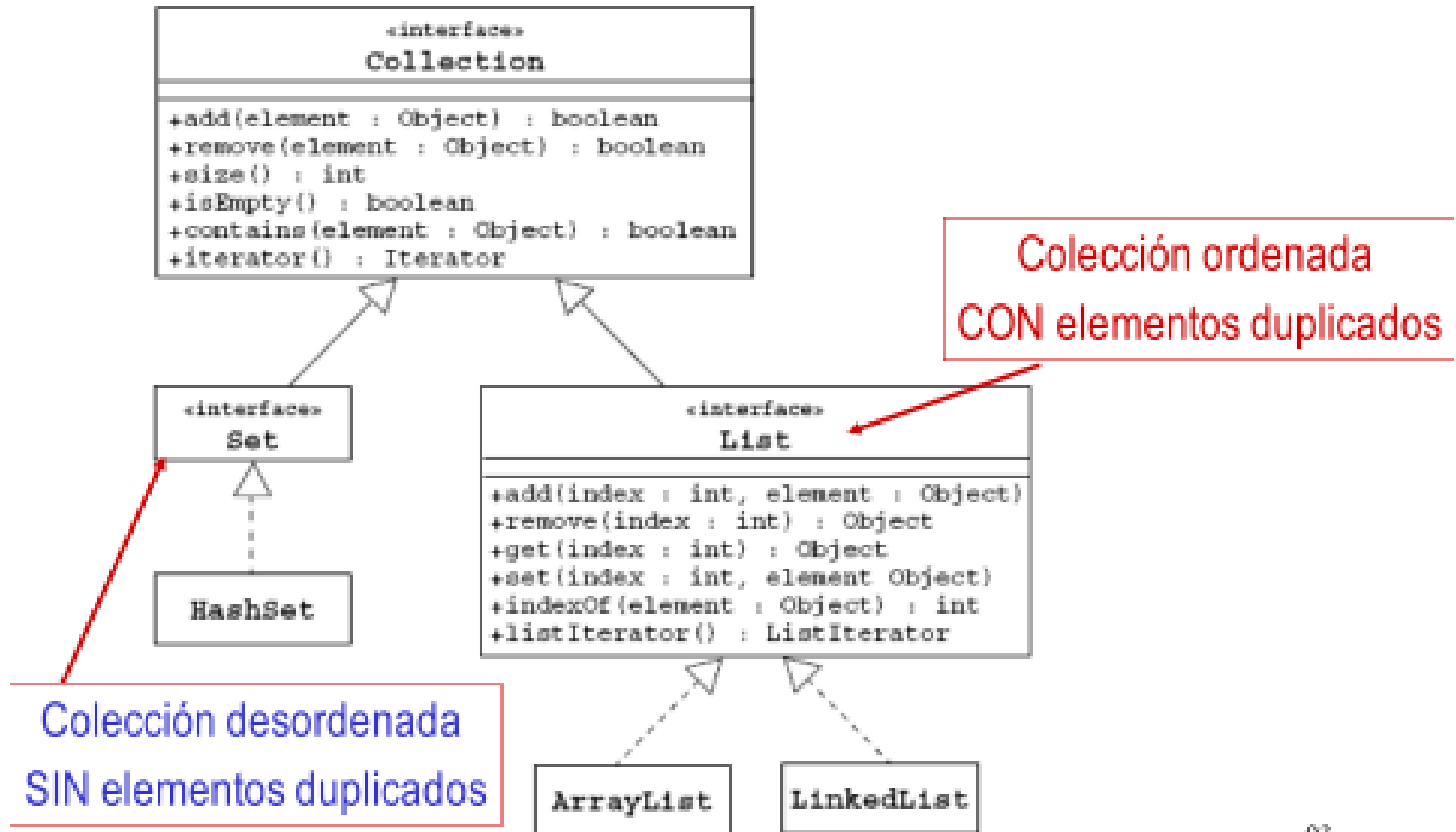
```
public class AgendaException extends Exception {  
  
    public AgendaException() {  
        super();  
    }  
  
    public AgendaException(String mensaje) {  
        super(mensaje);  
    }  
  
}
```

```
public Persona personaEnPosicion(int i) throws AgendaException{  
    Persona p = null;  
    if (i<0 || i>this.listaPersonas.size())  
        throw new AgendaException("Error: Indice no válido");  
  
    p=this.listaPersonas.get(i);  
    return p;  
}
```

Excepciones XI



The Collections API



Ejercicio



Ejercicio VIII de la guía



Muchas gracias!