Programación 2

Lenguaje Java - Manejo de Excepciones (*3era parte*)
Típos de Clases

Profesor: Eduardo Godoy.

eduardo.gl@gmail.com

Escuela de Ingeniería Civil Informática.
Universidad de Valparaíso.

4 de septiembre de 2017

Contenido

Manejo de Excepciones

- El termino Excepción significa Condición Excelcional dentro un programa.
- En java muchos evento requieren de manejo de Excepciones, Ejemplo:
  - Fallas en la comunicación con componentes de Hardware.
  - Operaciones con fuentes de almacenamiento persistente.
  - Operaciones aritméticas no permitidas.
  - etc.

Manejo de Excepciones: ¿Como Opera?

- El manejo de Excepciones opera transfiriendo el control del programa a cierta región del programa, siempre dentro del mismo método.

Ejemplo.

Si en la ejecución de un programa se encuentra una operación de división por 0, la ejecución del método no continua. Luego se pasa a ejecutar el código que ha sido prpeparado para manejar dicha excepción.

Manejo de Excepciones: try - catch

- Palabra reservada **Try** es usada para definir e identificar a un segmento de código en el cual puede ocurrir la excepción.
- Este bloque de código es llamado zona segura, debido a que ahí hay una o mas lineas riesgosas.
- Dentro de **catch** se indica el bloque de código que se encargara de controlar la excepción, Ejemplo:

Manejo de Excepciones - try, catch

Ejemplo.

```
try {
  Linea de codigo 1;
  Linea de codigo 2;
  Linea de codigo 3; <-- OCURRE EXCEPCION
  ...
  Linea de codigo n;
}catch(MiExcepcion){
  Codigo Manejo de excepcion 1;
  Codigo Manejo de excepcion 2;
  ...
  Codigo Manejo de excepcion 3;
}
```

Manejo de Excepciones: finally

- Palabra reservada **finally** es una sección que se ejecuta siempre que independiente si la excepción ocurra o no.
- Debido a que al ocurrir una excepción esta interrumpe la ejecución natural del código. Su principal función es mantener la consistencia y ejecución limpia del programa.

Manejo de Excepciones - try, catch, finally

Ejemplo.

```
try {
  Linea de codigo 1;
  Linea de codigo 2; <-- OCURRE EXCEPCION
  ...
  Linea de codigo n;
}catch(MiExcepcion){
  Codigo Manejo de excepcion 1;
  ...
  Codigo Manejo de excepcion n;
}finally{
  Codigo 1; ( se ejecutar\'a siempre)
  ...
  Codigo n;
}
```

Manejo de Excepciones: Declaración en métodos.

- Una excepción puede ser declarada en el método, con esto se deja la responsabilidad de controlar la excepcion a la clase que contiene al método que ha enviado el mensaje.

Manejo de Excepciones - Declaración en métodos

Ejemplo.

```
import java.io.*;
public class Test {
 public void myMethod1() {
  try{
     myMethod2();
  }catch(ArithmeticException ex){
    System.out.println("Division por Cero. Ocurrida en
        myMethod2");
  }finally{
    System.out.println("Siempre entro aqui");
  }
 }
```

Manejo de Excepciones - Declaración en métodos

Ejemplo.

```
 public int myMethod2() throws ArithmeticException  {
  float resultado = 1/0;
  return resultado;
 }
}
```

Final class

Consideraciones

- Proveer un mecanismo para cubrir un valor primitivo con características de objetos.
- En Java existe una clase Wrapper para todo dato primitivo.

Final class

Alugunos Wrappers

primitive	Wrapper class	Constructor Args.
boolean	Boolean	boolean o String
byte	Byte	byte o String
char	Char	char
double	Double	double o String
float	Float	float, double o String
int	Integer	int o String
long	Long	log Strin
short	Short	ceshort o String

Manejo de Excepciones - Declaración en métodos

Ejemplo.

```
Integer i1 = new Integer(42);
Integer i2 = new Integer("42");

Float f1 = new Float(3.14f);
Float f2 = new Float("3.14f");}

Character c1 = new Character('c');
```

API

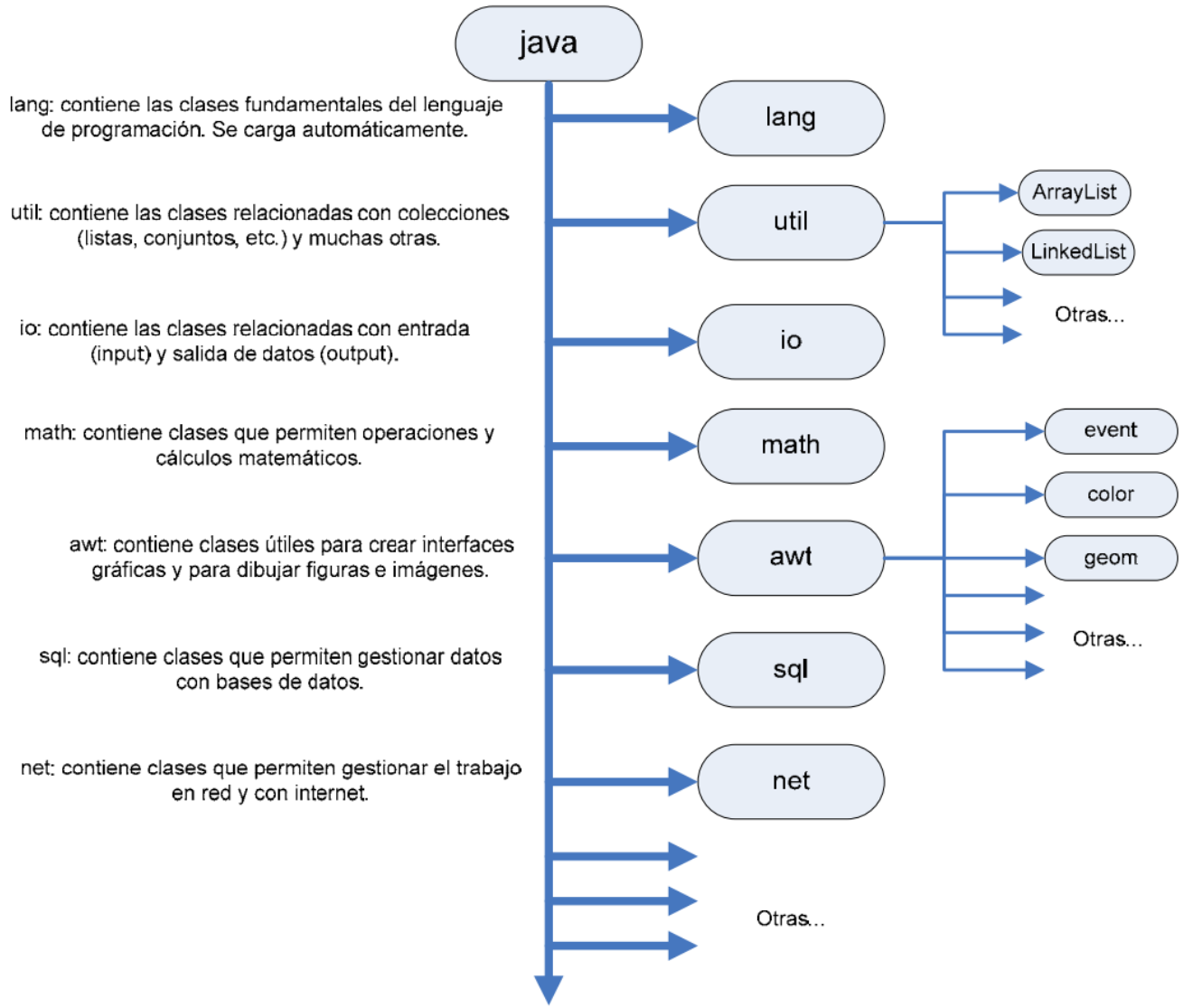
- **API** - *Application Programming Interface*.
- Es un conjunto de clases y métodos que permiten comunicarse con otras otros programas, componentes o servicios.
- La **API de Java** provee las funcionalidades necesarias para permitirnos utilizar sus características o ventajas sobre los otros lenguajes.
- Java cuenta con APIS para diversos objetivos, como por ejemplo, conectarse a un componente de Hardware, servicios en internet (sistemas de pago online), Comunicarse con tros lenguajes de programación, bases de datos, lenguajes de etiquetado (XML), etc.

API

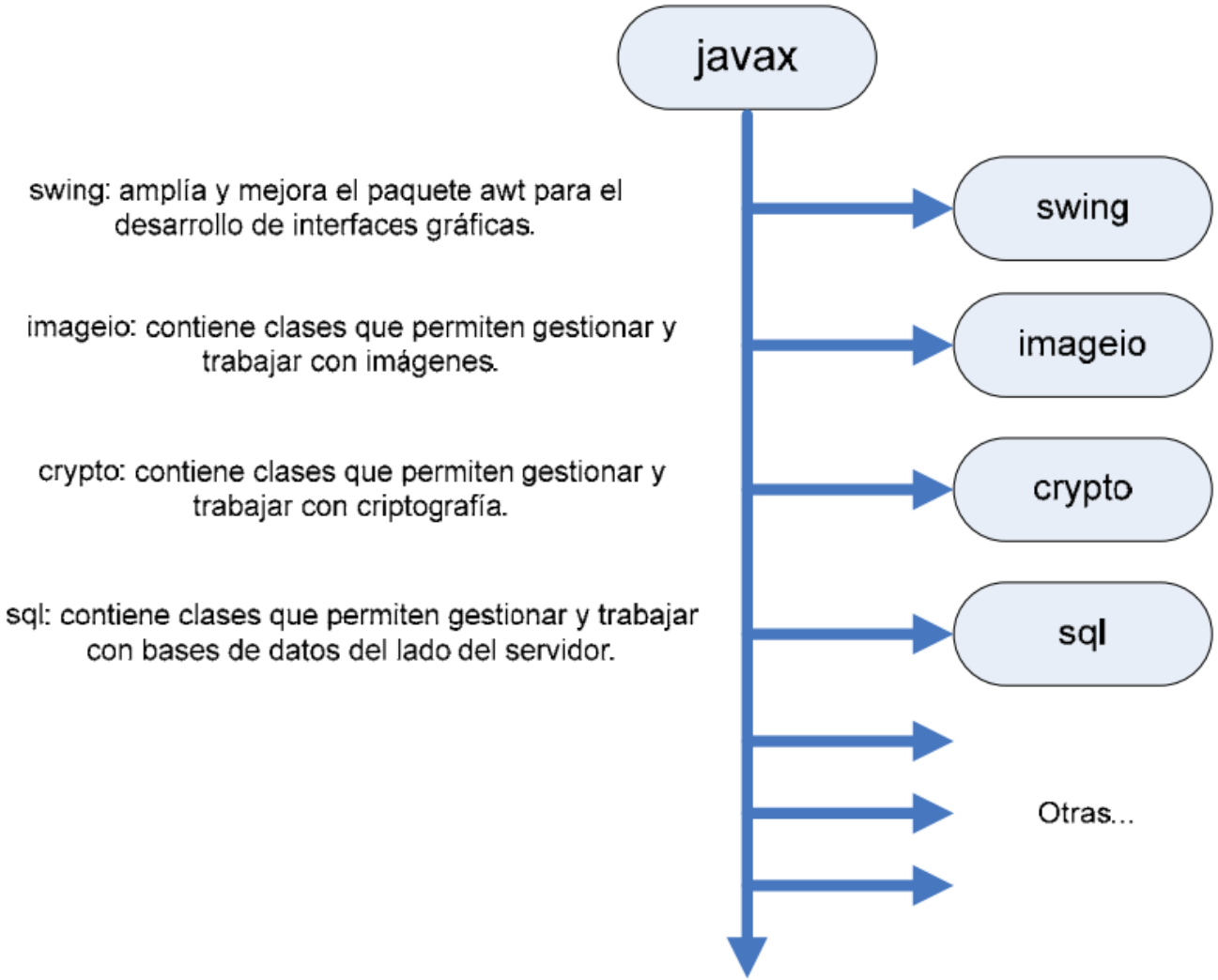
En general

Se conoce como API en java a un conjunto de librerias cuyos **imports** en nuestros programas nos ayudan a solucionar un problema de integración o comunicación específico

API - Ejemplo.


java
lang: contiene las clases fundamentales del lenguaje de programación. Se carga automáticamente.
lang
util: contiene las clases relacionadas con colecciones (listas, conjuntos, etc.) y muchas otras.
util
ArrayList
LinkedList
Otras...
io: contiene las clases relacionadas con entrada (input) y salida de datos (output).
io
math: contiene clases que permiten operaciones y cálculos matemáticos.
math
event
color
awt: contiene clases útiles para crear interfaces gráficas y para dibujar figuras e imágenes.
awt
geom
Otras...
sql: contiene clases que permiten gestionar datos con bases de datos.
sql
net: contiene clases que permiten gestionar el trabajo en red y con internet.
net
Otras...

API - Ejemplo.


javax
swing: amplía y mejora el paquete awt para el desarrollo de interfaces gráficas.
swing
imageio: contiene clases que permiten gestionar y trabajar con imágenes.
imageio
crypto: contiene clases que permiten gestionar y trabajar con criptografía.
crypto
sql: contiene clases que permiten gestionar y trabajar con bases de datos del lado del servidor.
sql
Otras...

Manejo de Excepciones - Declaración en métodos

Ejemplo.

```
import java.io.* //importando clases para entrada y
    salida (IN OUT)
import java.math.* //clases utiles para operaciones y
    valores matematicos
import java.util.*;
import java.util.ArrayList;

class EjemploImports{

}
```

Tips - Ocultamiento de Información - Ejemplo.

Visibility	Public	Protected	*Default*	Private
From the same class	Yes	Yes	Yes	Yes
From any class in the same package	Yes	Yes	Yes	No
From a subclass in the same package	Yes	Yes	Yes	No
From a subclass outside the same package	Yes	Yes, *through inheritance*	No	No
From any non-subclass class outside the package	Yes	No	No	No

Tips - Casting

Ejemplo.

```
class Animal {
  void makeNoise() {
    System.out.println("generic noise");
  }
}
class Dog extends Animal {
  void makeNoise() {
    System.out.println("woff");
  }
  void playDead() {
    System.out.println("aauuu");
  }
}
```

Tips - Casting

Ejemplo.

```
class MainAnimal {
  public static void main(String [] args) {
    Animal [] a = {new Animal(), new Dog(), new Animal()
        };
    for(Animal animal : a) {
      animal.makeNoise();
      if(animal instanceof Dog) {
        //animal.playDead(); //  que  pasa si descomento
            esto?
        Dog d = (Dog) animal;
        d.playDead();
      }
    }
  }
}
```