**TEMA 5**

**PROGRAMACIÓN ORIENTADA A OBJETOS**

El estado del objeto es el valor de los atributos, Es estado el objeto cambia a lo largo del programa,

Un objeto es una clase en ejecución

El ciclo de vida de un objeto es

Creación:

Uso:

Destrucción:

Las variables de tipo objeto se llaman referencias

Si ponemos directamente una variable sin valor (null) saldrá el error NullPointerException,

Para solucionarlo, habla que darle un valor

El **polimorfismo** es que el mismo método funcione de formas diferentes según el objeto que le llame

Los atributos tienen que ser PRIVADOS

GET para devolver el valor

SET para modificar

El encapsulamiento es la **ocultación de información**

El método constructor es NO ESTATICO, no tiene un valor de retorno, (Una clase por defecto tiene un constructor)

y tiene que ser público es un método que se ejecuta automáticamente para poner un estado inicial al objeto

Toda la clase object tiene los siguientes métodos:

* **toString**: Devuelve en forma de texto los valores de los atributos (no devuelve parámetros (El **@Override**, es un método que se está redefiniendo a otro método)
* **. equals ()**: Compara los atributos en memoria
* **getClass**: Sirve para saber cuál es la clase del objeto

COMPARACIONES ENTRE OBJETOS

* . toequals:
* . compareTo:

EXCEPCIONES

Una excepción es una condición anomala que ocurre mediante la ejecución del programa, deben ser manejadas por el programación, a proporcionar un código llamado **manejador de exapciones** (con **try catch**)