

Modelado y Diseño de Clases

2.3 Constructores/Destructores

Programación

Marzo 2018

Objetivo

- ▶ Comprender la importancia de los constructores y destructores en la POO

Constructores (1/2)

- ▶ En la POO los constructores son sub-rutinas que se ejecutan al crear un nuevo objeto
- ▶ Para crear un objeto es necesario reservar espacio en memoria e inicializar valores de los atributos que representan el estado del mismo objeto
- ▶ En general, los lenguajes de programación utilizan el operador `new` para la creación de objetos
- ▶ Los constructores tienen un nivel de acceso, un nombre (igual que la clase) y puede o no recibir parámetros
- ▶ Los constructores son elementos que no son heredados (independientemente de su nivel de acceso)

Constructores (2/2)

- ▶ En Java los constructores son “métodos especiales” que existen dentro de una clase, y son llamados de forma automática al momento de crear una instancia con **new**

- ▶ `Persona p = new Persona();`
- ▶ Un constructor por defecto es un constructor que no recibe parámetros y es invocado cada vez que se construye el objeto
- ▶ Al no inicializar ninguna propiedad de instancia estas serán inicializadas con valores predeterminados (0, “”, null, etc.)
- ▶ Constructor sin parámetros

```
class Ejemplo{  
    public Ejemplo() {}  
}
```

- ▶ Este constructor no es definido propiamente por el programador
- ▶ Al momento de definir de manera explícita un constructor el constructor por defecto es reemplazado por este

```
class Socio{  
    private String nombre;  
    public Socio() {} //Constructor sin parámetros, es recomendable colocarlo cuando se define explícitamente un constructor  
    public Socio(String nombre){  
        this.nombre = nombre;  
    }  
}
```

Destruyores

- ▶ En contraste con los constructores, los destruyores son “métodos especiales” que liberan el espacio de memoria de un objeto, con la intención de que ese espacio pueda ser ocupado nuevamente
- ▶ Las clases solo pueden tener un destructor y no aceptan modificadores de acceso y parámetros
- ▶ No pueden ser heredados ni sobrecargados
- ▶ Los destruyores no son llamados explícitamente, sino que son invocados automáticamente

- ▶ En lenguajes de programación como C++, o C# se emplea una sintaxis como:

```
~Persona()  
  
{  
    //Limpiar información, etc..  
}
```

- ▶ En Java no se definen destruyores, la liberación de los objetos es llevada a cabo por el Garbage Collector (recolector de basura) cuando las instancias de los objetos quedan sin referencias
- ▶ En Java el método dispose() de cada objeto se llama previo a ser recolectado