

Síntesis conceptual

Grado: Administración de sistemas informáticos en red
Asignatura: Programación básica
Unidad: 6. Clases y utilización de objetos

Resumen

Una clase es una representación abstracta de una entidad del mundo real en un programa informático que contiene unas propiedades y unas acciones.

Un objeto es una instancia creada en base a una clase. Es decir, una variable de esa clase contendrá las mismas propiedades y acciones definidas en la clase.

Una clase se puede declarar indicando el nivel de visibilidad, su nombre, el nombre de la clase de la que hereda, si no se indica nada por defecto se hereda de la superclase *Object* y las interfaces que pueda implementar.

Los campos, también denominados atributos, variables o fields. Permiten guardar la información que debe contener la clase. Por cada objeto instanciado de una clase contendrá valores diferentes en sus campos.

Los métodos dotan de acciones a la clase usando los datos de sus campos y se identifican por su nombramiento precedido por un verbo.

Existen diferentes tipos de métodos en las clases en Java:

- Constructores: Inician los objetos, pueden ser por defecto o con parámetros.
- Observadores: También mencionado como *getter*. Permiten consultar un campo.
- Modificadores: También mencionado como *setter*. Permiten modificar un campo.
- Métodos personalizados: Son métodos definidos según la funcionalidad que necesite la clase.

Dentro de los métodos podemos encontrar elementos como:

- Argumentos y parámetros: Cuando se invoque un método se tendrá que incluir todos los argumentos que hayamos definido como parámetro y que coincidan en el tipo.
- Valores de retorno: Los métodos pueden operar sin tener que devolver ningún valor o devolviendo algún resultado indicando *void* en la declaración del método.
- Sobrecarga de métodos: Declaración de un mismo método en una misma clase, pero con diferentes parámetros.

Los constructores son los encargados de inicializar los objetos de las clases. Para instanciar un objeto solo será necesario indicar el operador *new* junto al nombre de clase.

La encapsulación y visibilidad nos permite definir distintos niveles de protección al acceso de la información de una clase. La forma que tenemos para garantizar la protección en los datos es mediante la organización de paquetes y clases; y los modificadores de visibilidad en campos y métodos. Existen cuatro niveles de visibilidad:

- **Public:** Visibilidad menos restrictiva permite acceder a sus elementos desde el exterior y cualquier parte del programa.
- **Protected:** Se tendrá acceso a la clase desde el mismo paquete o subclases que estén en diferente paquete. Una clase externa en otro paquete no tendrá acceso.
- **Package o estándar:** Sin modificador definido. Se tendrá acceso a la clase desde el mismo paquete o subclases. Desde el exterior de otros paquetes u otras clases no se podrá.
- **Private:** La más restrictiva. Solo se puede acceder desde la misma clase no del exterior.

Los paquetes nos permiten agrupar según una misma idea varias clases de forma jerárquica. Es muy parecido al sistema de archivos de nuestro directorio. Los beneficios que otorga organizar nuestro programa en paquetes es la reutilización de código, organizar el proyecto, agrupamiento de clases y mayor seguridad.

Conceptos fundamentales

- **Recolector de basura:** Mecanismo de Java que se encarga de liberar la memoria de aquellos objetos que se han quedado sin referencia.
- **SWT:** La librería para componentes gráficos de java con la que fue hecha eclipse.
- **Static:** palabra reservada que convierte la visibilidad en un campo común entre todos los objetos de esa clase. Si un método necesitara usar un campo estático, obligatoriamente el método tiene que ser estático.
- **Java.io:** librería que contiene todas las clases para entrada y salida de programas hechos en java.
- **Apache commons:** Una librería con diferentes utilidades a la hora de escribir en ficheros, tratamiento de cadena de caracteres, etc.