Atributos y sus accesos

- Público: Los atributos pueden ser accedidos fuera de la clase sin ningún tipo de método, por el objeto correspondiente.

- Privado: Los atributos sólo pueden ser accedidos dentro de la misma clase por el objeto correspondiente

Métodos y sus accesos

- Público: Los métodos públicos pueden ser invocados desde cualquier clase por el objeto correspondiente

- Privado: Los métodos privados pueden ser invocados únicamente en la misma clase por el objeto correspondiente.

Si no asignamos los valores del constructor a un atributo, y en vez se los asignamos a una variable, lo tomara como *null*

Dentro del constructor aunque el nombre de un atributo y del parámetro es el mismo, no son la misma cosa

Los atributos por lo general tienen acceso privado, ya que no queremos que sean accedidos fuera de la clase.

Solo puede haber una clase pública en un archivo Java porque el nombre del archivo Java es el mismo que el nombre de la clase pública. Y obviamente no podemos tener un archivo con dos nombres diferentes.

Podemos crear un objeto de una clase en otra siempre y cuando el constructor tenga acceso *public* y ambos programas esten en el mismo directorio (paquete), también desde ese objeto podemos invocar a sus métodos y atributos siempre y cuando tengan acceso *public*, si son *private* Java no va a dejar acceder a ellos.

Podemos compilar todos los programas .java de un directorio a la vez, con el comando

$ javac *.java

Si queremos crear un objeto de una clase en otra, pero ambos están en diferentes directorios, tenemos que importar una clase en otra

El nombre de la clase, el archivo y el constructor deben ser el mismo.

Una buena práctica de programación es colocar cada clase en un archivo diferente, se pueden declarar clases dentro de otras clases en el mismo archivo pero no es lo ideal, ya que cuando son cientos o miles de líneas de código es desorganizado y poco óptimo

contexto *Static*

Lo utilizamos en un método para que el método no pueda ser invocado a partir de un objeto

imprimirBienvenida();

no podemos invocarlo así:

nombreObjeto.imprimirBienvenida();

Utilizamos las flechas en la terminal para acceder a los comandos anteriores