

Persistencia de datos (objetos)

En el caso de persistencia de objetos la información que persiste en la mayoría de los casos son los valores que contienen los atributos en ese momento, no necesariamente la funcionalidad que proveen sus métodos.

La persistencia de objetos puede ser fácilmente confundida con la persistencia en memoria; incluso con la persistencia de aplicación.

La persistencia de objetos consiste en la inicialización de objetos con sus atributos predeterminados o atributos por defecto. Esto es posible con dos maneras de proceder.

Sobre un medio (de almacenamiento) fijo se guarda (cuando el objeto fue definido) un conjunto de datos que son recuperados cuando el tipo de objeto en cuestión es creado; dichos datos son transferidos a las propiedades del objeto.

Otro objeto mantiene los datos que serán transferidos a las propiedades del nuevo objeto creado. En este caso los datos están en memoria.

Hay muchos ejemplos para este tipo de persistencia. Un caso típico son los **controles**. Cuando el control es compilado junto al código se guarda una copia de los datos que el programador definió de forma predeterminada. Cuando se instancia una referencia al control, este lee del disco (donde está almacenada la librería asociada al control) los datos que definen y configuran sus propiedades. Otro ejemplo son las primitivas que se utilizan para recrear objetos tridimensionales (en 3D) a partir de los cuales pueden crearse objetos más complejos. Supongamos que vamos a crear un avión con geometría 3D. Inicialmente podemos crear cada primitiva que compone el mismo, desde memoria, el motor de la aplicación puede definir un nuevo objeto con las medidas que el usuario define y posicionado igualmente en las coordenadas que el usuario señala para crear el objeto. Sin embargo, a medida que el objeto crece y tiene más datos, se hace necesario guardar el objeto en disco. Este objeto puede ahora ser guardado y ser utilizado las próximas veces como una primitiva, un punto a partir del cual iniciar la creación de otros objetos más complejos basados en este. Si ahora se quiere crear un nuevo modelo de avión posiblemente sea más sencillo modificar la primitiva definida que crear uno totalmente nuevo; por tanto, es necesaria la persistencia de la primitiva para recrear nuevos objetos basados en ella. Las primitivas complejas se acoplan a la primera manera definida y las primitivas sencillas se ajustan a la segunda manera definida.

Para guardar los datos de objetos en disco se recurre a un mecanismo conocido como serialización (serializar), que dispone en una secuencia de bytes todos los datos (o sólo aquellos que se desee) que definen el objeto.

Desde la óptica de la persistencia, se podrían clasificar los objetos en:

Transitorios: cuyo tiempo de vida depende directamente del ámbito del proceso que los instanció.

Persistentes: cuyo estado es almacenado en un medio secundario para su posterior reconstrucción y utilización, por lo que su tiempo de vida es independiente del proceso que los instanció.

Al programador, la persistencia permite almacenar, transferir y recuperar el estado de los objetos. Para esto existen varias técnicas:

- Serialización,
- Motores de persistencia,
- Bases de datos orientadas a objetos.