**GENERIC CLASSES AND TYPE PARAMETERS**

* La clase LinkedList (Creada manualmente en el libro) es un ejemplo de programación genérica, podemos guardar objetos de cualquier clase dentro de una LinkedList
* La clase LinkedList logra la genericidad mediante el uso de herencia. Utiliza una referencia del tipo Object y por lo tanto es capaz de guardar objetos de cualquier clase.
* Por ejemplo, podemos agregar elementos de tipo String porque la clase String extiende de la clase Object.

**En java, la programación genérica puede ser lograda con herencia o con parámetros de tipo.**

* En contraste, las clases ArrayList y LinkedList de la librería estándar de Java son clases genéricas.
* Cada una de estas clases tiene un **parámetro de tipo (type parameter)** para especificar el tipo de sus elementos.
* Por ejemplo: ArrayList<String> guarda elementos de tipo String.

**Una clase genérica tiene uno o dos parámetros de tipo.**

* Cuando declaramos una clase genérica, debemos indicar una variable para cada **parámetro de tipo**. Por ejemplo, la librería estándar de java declara la clase **ArrayList<E>,** donde E es la **variable de tipo (type variable)** que denota el tipo de elemento.
* Usamos la misma variable en la declaración de los métodos cuando se necesario referirse a ese tipo. Por ejemplo, la clase **ArrayList<E>** contiene dos métodos:

|  |
| --- |
| **public void add(E element)** |
| **public E get(int index)** |

* Se puede utilizar otro nombre, como ***TipoDeElemento*** en lugar de **E**. Sin embargo, es costumbre utilizar nombres de una sola letra mayúscula para las variables de tipo.
* Al utilizar una clase genérica, necesitamos instanciar el parámetro de tipo. Esto es, proporcionar algún tipo de dato. Podemos proveer cualquier tipo de clase o interface. Por ejemplo:

|  |
| --- |
| **ArrayList<CuentaBanco>** |
| **ArrayList<Measurable>** |

**Los parámetros de tipo pueden ser instanciados con clases o interfaces.**

* No podemos sustituir o utilizar alguno de los ocho tipos primitivos de datos. Sería un error declarar ArrayList<double>. En lugar de eso se utiliza la clase wrapper, ArrayList<Double>
* Cuando instanciamos una clase genérica, el parámetro de tipo que proporcionamos remplaza todas las ocurrencias de la variable de tipo en la declaración de la clase.
* Por ejemplo, el método **add** para **ArrayList<CuentaBanco>** tiene la variable de tipo **E** que es remplazada con **CuentaBanco**

|  |
| --- |
| **public void add(E element)** |
| **Public void add(CuentaBanco element)** |