APRENDERAPROGRAMAR.COM

CONCEPTO O DEFINICIÓN DE POLIMORFISMO EN JAVA. ¿QUÉ ES EL POLIMORFISMO? APROXIMACIÓN. EJEMPLO PRÁCTICO. (CU00678B)

Sección: Cursos
Categoría: Curso “Aprender programación Java desde cero”

Fecha revisión: 2029

Resumen: Entrega nº78 curso Aprender programación Java desde cero.

Autor: Alex Rodríguez

CONCEPTO DE POLIMORFISMO EN JAVA. ¿QUÉ ES?

En el anterior apartado del curso hemos indicado que es posible denominar “edificio a dos aguas” a una capilla que cumpla la norma, también a una vivienda unifamiliar que la cumpla, o a una biblioteca que la cumpla. Y que podemos decir que son List las clases que cumplen con la interface como ArrayList, LinkedList y otras.

Fijémonos ahora en estas formas de sintaxis:

FORMA 1	FORMA 2
private ArrayList <Integer> serieAleatoria; … ejemplo aprenderaprogramar.com serieAleatoria = new ArrayList <Integer> ();	private List <Integer> serieAleatoria; … ejemplo aprenderaprogramar.com serieAleatoria = new ArrayList<Integer> ();
Aquí se construirá un edificio a dos aguas tipo capilla; … ejemplo aprenderaprogramar.com Construimos un edificio a dos aguas tipo capilla;	Aquí se construirá un edificio a dos aguas; … ejemplo aprenderaprogramar.com Construimos un edificio a dos aguas tipo capilla;

Lo que estamos comprobando aquí es que Java permite declarar variables sin especificar la clase concreta a la que van a pertenecer, sino únicamente diciendo qué interface van a cumplir. Por este motivo decimos que **una interface define un tipo**. Esto explica a su vez por qué en la documentación del API de Java nos encontramos ocasiones en que vemos que un método requiere como parámetro un tipo que no se corresponde con una clase, sino con una interface. Supongamos que un método pide como parámetro un tipo List. Eso significa que admite que se le pase como parámetro tanto un ArrayList como un LinkedList como otro objeto que cumpla con la interface. En este caso diríamos que el parámetro es polimórfico porque admite distintas formas de objeto, no solo una. El polimorfismo se manifiesta de distintas formas en Java como iremos viendo en los próximos apartados. Trata de razonar sobre el significado del siguiente código:

```
//Ejemplo aprenderaprogramar.com
ArrayList <List> misListas = new ArrayList <List> ();
ArrayList<Integer> miListaIntegers = new ArrayList<Integer>();
LinkedList<String> miListaStrings = new LinkedList<String>();
misListas.add (miListaIntegers);
misListas.add (miListaStrings);
```

Con los conocimientos explicados hasta el momento debemos ser capaces de interpretar estas líneas. En primer lugar declaramos un ArrayList que va a contener objetos de tipo List: eso significa que dentro del ArrayList podrá haber ArrayList, LinkedList, Stack, etc. A continuación creamos un ArrayList de integers y un LinkedList de strings. Y finalmente, añadimos esos objetos a misListas. Esta variable decimos que es polimórfica porque contiene objetos que no son solo de un tipo, sino de varios tipos.

EJERCICIO

La interface Set de Java es implementada por las clases HashSet y TreeSet. Busca información sobre estas clases en la documentación del api Java. Crea un programa Java que haga lo siguiente:

a) Declarar un ArrayList de objetos de tipo Set

b) Crear un objeto de tipo HashSet para contener Strings y haz que contenga las cadenas “sol”, “luna”, “saturno”.

c) Crear un objeto TreeSet para contener Integers y haz que contenga los números 2, 8, 5.

d) Añade los objetos HashSet y TreeSet como elementos del ArrayList.

e) Usa iteradores para recorrer los elementos del ArrayList y recorrer el contenido de cada uno de los elementos y mostrar este contenido por pantalla. Por pantalla deberás obtener “sol”, “luna”, “saturno”, 8, 5, 2.

Puedes comprobar si tu código es correcto consultando en los foros aprenderaprogramar.com.

Próxima entrega: CU00679B

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:

http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=68&Itemid=188