

UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO
Facultad de Ciencias



Introducción a las Ciencias de la Computación

Práctica 11: Herencia

Profesora:

Amparo López Gaona

Ayudante:

Víctor Emiliano Cruz Hernández

Ayudante de Laboratorio:

Kevin Jair Torres Valencia

Objetivos

El objetivo de esta práctica es que el alumno refuerce sus conocimientos acerca del paso de parámetros por valor, la diferencia entre parámetro formal y parámetro real o actual, así como el paso de parámetros al método main. Que además ejercite en la creación de jerarquías de herencia de clases, ya sea de la raíz a las hojas o de las hojas a la raíz.

Introducción

La herencia permite definir una nueva clase C_n a partir de una clase existente C , muy parecida a C_n , evitando la duplicidad de código. En la clase C_n se definen sólo los atributos y los métodos que difieren de los existentes en la clase C y automáticamente se incluyen los métodos y atributos de C . Como resultado, los atributos y métodos de C_n son todos los de C más los especificados en C_n .

Estos últimos pueden ser nuevos métodos o bien la redefinición o cancelación de métodos ya existentes en C . La clase C_n se denomina subclase o clase derivada y la clase C se conoce como superclase, clase base o clase padre.

Para especificar que se va usar el mecanismo de herencia se utiliza la palabra reservada **extends** seguida del nombre de la clase que se heredará.

Los objetos de una subclase son como los objetos de cualquier otra clase, por lo tanto los elementos públicos de su superclase y de cualquier otra clase pueden ser usados en ella sin ningún problema, sin embargo, los atributos y métodos privados no pueden ser usados fuera de la clase en que se definieron, así sea una subclase. Si se quiere tener atributos y métodos que sean privados para todas las clases, excepto para las clases derivadas, es necesario que en su declaración se precedan de la palabra reservada **protected**.

Como se sabe, al crear un objeto se llama a un constructor de su clase para asegurar que se cree con un estado inicial válido. Cuando se crea un objeto de una clase derivada implícitamente se crea un objeto de la clase base. Al programar una subclase es necesario programar algún método constructor, el cual, en general, incluye la llamada a algún constructor de la **superclase**; esto se hace mediante la instrucción **super** con los argumentos adecuados.

La llamada al constructor de la superclase debe ser la primera instrucción del constructor de la subclase, con lo cual se empieza por asignar un estado inicial a la parte heredada y luego se inicializa la parte propia de la clase.

Desarrollo

1. Escribir un programa que permita a una persona llevar el control de los libros, discos (cd) y películas que posee. La información que le interesa de cada uno sus artículos es:

1. Libro:

- Título.
- Autor.
- Editorial.
- Año de publicación.

2. Disco:

- Título.
- Género.
- Intérprete.
- Número de canciones.
- Idioma.
- Año de lanzamiento.

3. Película:

- Título.
- Director.
- Género.
- Duración.
- Año de filmación.
- Idioma.

Debes crear las clases necesarias para resolver el problema planteado, como mínimo serán tres: `Libro`, `Disco`, `Pelicula`. Estas clases deben de heredar de la clase `Articulo.class`. Cada clase debe programarse en un archivo con el nombre de la misma. En cada clase se debe incluir:

- Un método constructor que reciba los parámetros necesarios para dar valor inicial a los atributos de sus objetos.
- Métodos modificadores e inspectores para cada uno de los atributos que tendrán sus objetos. No se deben duplicar métodos.
- Un método `toString` para recuperar el objeto de esa clase en forma de cadena.

- Documentación de cada método. Para ello utilizar etiquetas `@author`, `@version`, `@param` y `@return`.

Por último debes escribir un programa `PruebaArticulos` para comprobar que la jerarquía de clases se construyó correctamente. Este programa debe trabajar a través de un menú de opciones que permita ingresar la información de los artículos e imprimir por separado libros, discos y películas almacenados en el programa.

Formato de Entrega

1. Las prácticas se entregarán en parejas.
2. Cada práctica (sus archivos y directorios) deberá estar contenida en un directorio llamado EquipoX_pY, donde:
 - (a) X es el número de equipo correspondiente.
 - (b) Y es el número de la práctica.

Por ejemplo: Equipo09_p01

3. NO incluir los archivos .class dentro de la carpeta.
4. Los archivos de código fuente deben estar documentados.
5. Se pueden discutir y resolver dudas entre los integrantes del grupo. Pero cualquier práctica plagiada total o parcialmente será penalizada con cero para los involucrados.
6. La práctica se debe subir al Github Classroom correspondiente.
7. La entrega en classroom debe contener el link HTTPS y SSH de su repositorio y es lo único que se debe entregar.
8. El horario y día de entrega se acordará en la clase de laboratorio y no deberá sobrepasar 2 clases de laboratorio.