

UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO
Facultad de Ciencias



Introducción a las Ciencias de la Computación

Práctica 09: Parámetros y herencia.

Profesora: Amparo López Gaona
Ayudante: Ramsés Antonio López Soto
Ayudante: Adrián Aguilera Moreno

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.

Con frecuencia se tiene que los métodos requieren de algunos datos para realizar su tarea, estos datos se conocen como parámetros. Los parámetros formales se especifican, después del nombre del método, entre paréntesis, como una lista de parejas separadas por comas. Cada pareja incluye el tipo y el nombre de cada parámetro. Los paréntesis son parte de la sintaxis,

así que deben estar presentes aunque el método no requiera parámetros. Existen diferencias entre la declaración de una variable y la de un parámetro. En los parámetros no se especifica la visibilidad, cada parámetro se precede de su tipo y la definición no termina con punto y coma. El dato con el que se llama a ejecutar el método se conoce como parámetro real o bien como parámetro actual. Al llamar a ejecución un método, el valor del parámetro real se asigna como valor inicial del parámetro formal y termina la relación entre ambos parámetros; es decir, si en el cuerpo del método se modifica el valor del parámetro formal no cambia el valor del parámetro real; esto se conoce como paso de parámetros por valor.

El método `main` tiene como parámetro un arreglo de cadenas cuyos valores se proporcionan al llamar a ejecutar el programa. Éstos son cadenas separadas por espacios en blanco. Si se desea que alguna cadena incluya un espacio debe encerrarse ésta entre comillas. Una vez en ejecución el método `main`, su parámetro puede usarse como cualquier otro arreglo de cadenas.

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.

2. Escribir programas cuyo método `main` reciba una serie de parámetros para poder trabajar dando diferentes resultados de acuerdo con dichos parámetros.

1. Escribir un programa `Nombre` que reciba `n` parametros que se supone son el nombre completo de una persona y los escriba empezando por los apellidos.

Ejemplos:

```
>java Nombre Juan Paco Pedro "de la Mar" Perez
El nombre completo es: de la Mar Perez Juan Paco Pedro.
El nombre es: Juan Paco Pedro.
```

```
>java Nombre
Debes ingresar al menos tres cadenas para el nombre.
```

```
>java Nombre Ana Maria
Debes ingresar al menos tres cadenas para el nombre.
```

```
>java Nombre Ana Patricia Maria
El nombre completo es: Patricia Maria Ana
El nombre es: Ana
```

2. Escribir un programa `Numeros` que lea 4 cadenas, la primera indica qué hacer y las otras tres deben tratarse como números.

Los posibles valores para la primera cadena son un guión seguido de alguna de las siguientes letras M, m, p,o, t. El significado es:

- M determina el valor mayor.
- m determina el valor menor.
- p calcula el promedio de los tres valores.
- o muestra los tres valores en orden decreciente.

- t realiza las cuatro funciones anteriores.

Ejemplos de ejecución del programa son los siguientes:

```
> java Numeros -M 45 78 90
El mayor de 45, 78 y 90 es 90
```

```
> java Numeros -p 45 345 90
El promedio de 45, 345 y 90 es 160
```

```
> java Numeros -t 45 345 90
El mayor de 45, 345 y 90 es 345
El menor de 45, 345 y 90 es 45
El promedio de 45, 345 y 90 es 160
Los números ordenados son 345, 90, 45
```

```
> java Numeros -x 45 345 90
Opcion incorrecta !!!
```

```
> java Numeros -t 45
Cantidad incorrecta de datos!!!
```

Formato de Entrega

1. Las prácticas serán entregadas de forma individual.
2. Cada práctica (sus archivos y directorios) deberá estar contenida en un directorio llamado `apellidoPaterno_nombre_pX`, donde X es el número de la práctica.
Por ejemplo: `aguilera_adrian_p9`
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.