Trabajo Práctico de Laboratorio de Programación

Instituto Agustín Elizalde.

di Nápoli, Franco.

6°Técnica.

2018.

CONSIGNAS

1) ¿Qué es POO?

2) ¿Cuál es su ventaja por sobre la programación estructurada?

3) ¿Qué es una clase?

4) ¿Qué es un objeto?

5) Definir método y atributos.

6) ¿Qué son los setters? ¿Para qué sirven?

7) ¿Qué son los getters? ¿Para qué sirven?

8) ¿Qué es instanciar un objeto?

9) ¿Qué es un constructor? Diferenciar los diferentes tipos

10) Netbeans es un IDE de Software Libre, ¿Qué significa esto?

1) La programación orientada a objetos es un paradigma de programación que viene a innovar la forma de obtener resultados. Los objetos manipulan los datos de entrada para la obtención de datos de salida específicos, donde cada objeto ofrece una funcionalidad especial.  
Muchos de los objetos prediseñados de los lenguajes de programación actuales permiten la agrupación en bibliotecas o librerías, sin embargo, muchos de estos lenguajes permiten al usuario la creación de sus propias bibliotecas.  
Está basada en varias técnicas, incluyendo herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento.  
Su uso se popularizó a principios de la década de 1990. En la actualidad, existe una gran variedad de lenguajes de programación que soportan la orientación a objetos.

Algunos lenguajes orientados a objetos pueden ser:

* C++
* C#
* Java
* Python
* Objective-C
* Swift
* Ruby
* JavaScript
* Dart

2) Sus principales ventajas son:

* Reusabilidad: Cuando hemos diseñado adecuadamente las clases, se pueden usar en distintas partes del programa y en numerosos proyectos.
* Mantenibilidad: Debido a las sencillez para abstraer el problema, los programas orientados a objetos son más fáciles de leer y entender, nos permiten ocultar detalles de implementación dejando visibles sólo aquellos detalles más relevantes.
* Modificabilidad: La facilidad de añadir, suprimir o modificar nuevos objetos nos permite hacer modificaciones de una forma muy sencilla.
* Fiabilidad: Al dividir el problema en partes más pequeñas podemos probarlas de manera independiente y aislar mucho más fácilmente los posibles errores que puedan surgir.
* La facilidad de creación de programas.
* La agilización del desarrollo del software.
* La facilita el mantenimiento del software.
* La facilidad a la hora de crear nuevos tipos de objetos a partir de los ya existentes

3) Una clase es una plantilla para la creación de objetos de datos según un modelo predefinido. Las clases se utilizan para representar entidades o conceptos, como los sustantivos en el lenguaje. Cada clase es un modelo que define un conjunto de variables -el estado, y métodos apropiados para operar con dichos datos -el comportamiento. Cada objeto creado a partir de la clase se denomina instancia de la clase.

Ejemplo de clase en Java:

Class Movie{

//Atributos

public int duracion;

public String titulo;

public String autor;

public boolean empezar = false;

//Constructor

Movie(int d,String t,String a){

this.duracion = d;

this.titulo = t;

this.autor = a;

}

//Metodos

public void comenzar(){

empezar = true;

}

public void pausar(){

empezar = false;

}

}

4) Un objeto es una unidad dentro de un programa de computadores que consta de un estado y de un comportamiento, que a su vez constan respectivamente de datos almacenados y de tareas realizables durante el tiempo de ejecución. Un objeto puede ser creado instanciando una clase, como ocurre en la programación orientada a objetos, o mediante escritura directa de código y la replicación de otros objetos, como ocurre en la programación basada en prototipos.

Estos objetos interactúan unos con otros, en contraposición a la visión tradicional en la cual un programa es una colección de subrutinas (funciones o procedimientos), o simplemente una lista de instrucciones para el computador. Cada objeto es capaz de recibir mensajes, procesar datos y enviar mensajes a otros objetos de manera similar a un servicio.

En el mundo de la programación orientada a objetos (POO), un objeto es el resultado de la instanciación de una clase.

5) Un método es una subrutina cuyo código es definido en una clase y puede pertenecer tanto a una clase, como es el caso de los métodos de clase o estáticos, como a un objeto, como es el caso de los métodos de instancia. Análogamente a los procedimientos en lenguajes imperativos, un método consiste generalmente de una serie de sentencias para llevar a cabo una acción, un juego de parámetros de entrada que regularán dicha acción o, posiblemente, un valor de salida (o valor de retorno) de algún tipo.

La diferencia entre un procedimiento (generalmente llamado función si devuelve un valor) y un método es que este último, al estar asociado con un objeto o clase en particular, puede acceder y modificar los datos privados del objeto correspondiente de forma tal que sea consistente con el comportamiento deseado para el mismo. Así, es recomendable entender a un método no como una secuencia de instrucciones sino como la forma en que el objeto es útil (el método para hacer su trabajo). Por lo tanto, podemos considerar al método como el pedido a un objeto para que realice una tarea determinada o como la vía para enviar un mensaje al objeto y que éste reaccione acorde a dicho mensaje

Los atributos son las características individuales que diferencian un objeto de otro y determinan su apariencia, estado u otras cualidades. Los atributos se guardan en variables denominadas de instancia, y cada objeto particular puede tener valores distintos para estas variables.

Las variables de instancia también denominados miembros dato, son declaradas en la clase pero sus valores son fijados y cambiados en el objeto.

Además de las variables de instancia hay variables de clase, las cuales se aplican a la clase y a todas sus instancias. Por ejemplo, el número de ruedas de un automóvil es el mismo cuatro, para todos los automóviles.

En el punto 3 con la clase ‘Movie’ se ve el ejemplo los atributos y los métodos.

6-7) Los Setters y Getters son métodos de acceso lo que indica que son siempre declarados públicos, y nos sirven para dos cosas:

Setters: Del Inglés Set, que significa establecer, pues nos sirve para asignar un valor inicial a un atributo, pero de forma explícita, además el Setter nunca retorna nada (Siempre es void), y solo nos permite dar acceso público a ciertos atributos que deseemos el usuario pueda modificar.

Getters: Del Inglés Get, que significa obtener, pues nos sirve para obtener (recuperar o acceder) el valor ya asignado a un atributo y utilizarlo para cierto método.

Ejemplo en Java:

Class Persona{

public String nombre;

public String apellido;

public void setNombre (String n){

nombre = n;

}

public String getNombre(){

return nombre;

}

public void setApellido (String a){

apellido = a;

}

public String getApellido(){

return apellido;

}

}

8-9)

Un constructor es un método especial de una clase que se llama automáticamente siempre que se declara un objeto de esa clase.  
Su función es inicializar el objeto y sirve para asegurarnos que los objetos siempre contengan valores válidos.  
  
Cuando se crea un objeto en Java se realizan las siguientes operaciones de forma automática:  
1. Se asigna memoria para el objeto.  
2. Se inicializan los atributos de ese objeto con los valores predeterminados por el sistema.  
3. Se llama al constructor de la clase que puede ser uno entre varios.  
El constructor de una clase tiene las siguientes características:  
  
Tiene el mismo nombre que la clase a la que pertenece.  
  
En una clase puede haber varios constructores con el mismo nombre y distinto número de argumentos (se puede sobrecargar)  
  
No se hereda.  
  
No puede devolver ningún valor (incluyendo void).

Tipos:

Constructor por defecto: Un constructor por defecto es un constructor sin parámetros que no hace nada. Sin embargo será invocado cada vez que se construya un objeto sin especificar ningún argumento, en cuyo caso el objeto será iniciado con los valores predeterminados por el sistema (los atributos numéricos a cero, los alfanuméricos a nuloPeq, y las referencias a objetos a null).

El accessSpecifier de un constructor es el mismo que el de las variables. El nombre del constructor debe coincidir con el nombre de la clase. Por lo tanto, si llama a su clase Person, entonces el nombre del constructor también debe ser Person.

Para cualquier constructor que no sea el constructor predeterminado, usted pasa una argumentList, que es una o más de:

argumentType argumentName

Los argumentos en una argumentList se separan con comas y dos argumentos no pueden tener el mismo nombre. El argumentType es un tipo primitivo u otro tipo de clase (lo mismo que sucede con los tipos de variables).

La instanciación de un objeto se obtiene llamando al constructor de la clase y asignándole valores a los atributos de dicha clase.

Ejemplo de instanciación de un objeto (Clase Movie ejercicio 3):

Movie m = new Movie(140,”Harry Potter”, “JK Rowling”);

10) Que NetBeans IDE sea un software libre significa que el usuario o empresa que lo utilice puede modificar su código fuente para distintos usos específicos que desee.