

DISEÑO E IMPLEMENTACIÓN DE SOLUCIONES

AUTOR: WALTER MADRIGAL CHAVES

NOVIEMBRE: 2020



San Marcos

Contenido

INTRODUCCIÓN.....	2
PROGRAMACIÓN ORIENTADO A OBJETOS	3
OBJETOS.....	3
CLASES	4
ATRIBUTOS DE UNA CLASE	6
MÉTODOS	7
CONSTRUCTORES:.....	8
DESTRUCTOR.....	9
CONCLUSIONES Y RECOMENDACIONES	10
REFERENCIAS BIBLIOGRÁFICAS.....	11



INTRODUCCIÓN

La Programación orientada a objetos (POO) es un paradigma en el desarrollo de aplicaciones que engrandece su potencial con la posibilidad de construir clases, métodos y eventos. Esta es una cultura que respeta la programación como un generador de soluciones de problemas a gran escala, que requiere un análisis cuidadoso y una división de los problemas en estructuras manejables.

La POO es la piedra angular de los futuros desarrollos hacia la reutilización del código y es clave para la programación moderna, a diferencia de los otros modelos de programación como el assembler, programación por módulos o procedimientos, funcional, por eventos y otros.

Al comparar la POO con la programación de procedimientos y realizar un examen superficial del código escrito en ambos estilos, muestra que el código orientado a objetos tiende a descomponerse en un gran número de estructuras pequeñas, para luego ser verificado en un proceso de compilación.

Conocer conceptos, técnicas, métodos y buenas prácticas de la POO facilita el análisis y consecución de un código apropiado para hallar soluciones bien implementadas a los futuros desarrolladores de software. Se desarrollarán los conceptos de POO, las metas y objetivos de esta programación con la utilización de algunos IDE, se aprenderá acerca de las clases, atributos y métodos, y se aplicará técnicas como encapsulamiento y ocultamiento de la información.

PROGRAMACIÓN ORIENTADO A OBJETOS

Los diferentes tipos de desarrollo como de aplicaciones, sistemas de información, programas a la medida, entre otros, convergen en que su concepción debe ser cuidadosa, detallada y basada en una ideal real y concreta. Adoptar un modelo de programación, explotarlo y dominarlo sirve para escalar a otros paradigmas en la programación y poder reconocer cuál es más apto para aplicar en un determinado desarrollo.

La programación orientada a objetos es una técnica de desarrollo de sistemas muy novedosa en la forma de obtener resultados. Como su nombre lo indica, su lógica descansa en la creación de objetos, métodos y atributos, dándole a cada objeto una funcionalidad especial.

OBJETOS

Los objetos son representaciones de cosas, pueden ser simples o complejas y reales o imaginarias, ejemplos: casa, automóvil y flor. No todo puede ser considerado como un objeto, algunas cosas son simplemente características o atributos de los objetos: color, velocidad, nombre.

En el desarrollo de sistemas el término de objeto se refiere a una forma de agrupar un conjunto de datos y de funcionalidades en un mismo bloque de código, para luego ser referenciados desde otras partes del programa.

Se trata de un ente abstracto que permite separar los diferentes componentes de un programa, simplificando así su elaboración, depuración y posteriores mejoras. Los objetos integran, a diferencia de los métodos procedurales, tanto los procedimientos como las variables y datos referentes al objeto.

El siguiente es un ejemplo de la declaración de un objeto en Java

Imagen 1 Declaración de tres objetos

```

public static void main (String[] args){
    Coche miCoche = new Coche ("verde", 80, 3.2f);
    Coche tuCoche = new Coche ("rojo", 120, 4.1f);
    Coche suCoche = new Coche ("amarillo", 100, 3.4f);
}
  
```

Fuente: Elaboración propia

Los objetos se componen de 3 partes fundamentales: métodos, eventos y atributos. Además, el objeto tiene una identidad, un tipo y un valor. Una identidad de objeto nunca cambia una vez es creada.

El tipo de un objeto también es inmutable. El tipo de un objeto determina las operaciones que admite el objeto (por ejemplo, “¿tiene una longitud?”) Y también define los valores posibles para los objetos de ese tipo. Por último, El valor, que es un dato variable. Se dice que los objetos cuyo valor puede cambiar son mutables; los objetos cuyo valor no se puede cambiar una vez que se crean se llaman inmutables.

Para ahondar más en el tema de objetos en POO debe realizar la lectura de las páginas 31 a la 44 del libro: Programación orientada a objetos con Java usando Bluej 6A. ED. (2017) de David J. Barnes y Micahel Kölling.

CLASES

Las clases son un pilar fundamental dentro de la programación orientada a objetos. La mayoría de los lenguajes modernos incluyen la posibilidad de usar clases y objetos, convirtiéndose en las herramientas más utilizadas de la era

moderna.

Una clase se compone de un conjunto de objetos similares, es mediante los métodos y los datos que muestran las características comunes de dicho conjunto. Expresado de otra manera, una clase es la declaración de un tipo de objeto. Las clases son similares a los tipos abstractos de datos y equivalen a modelos que describen cómo se construyen ciertos tipos de objetos.

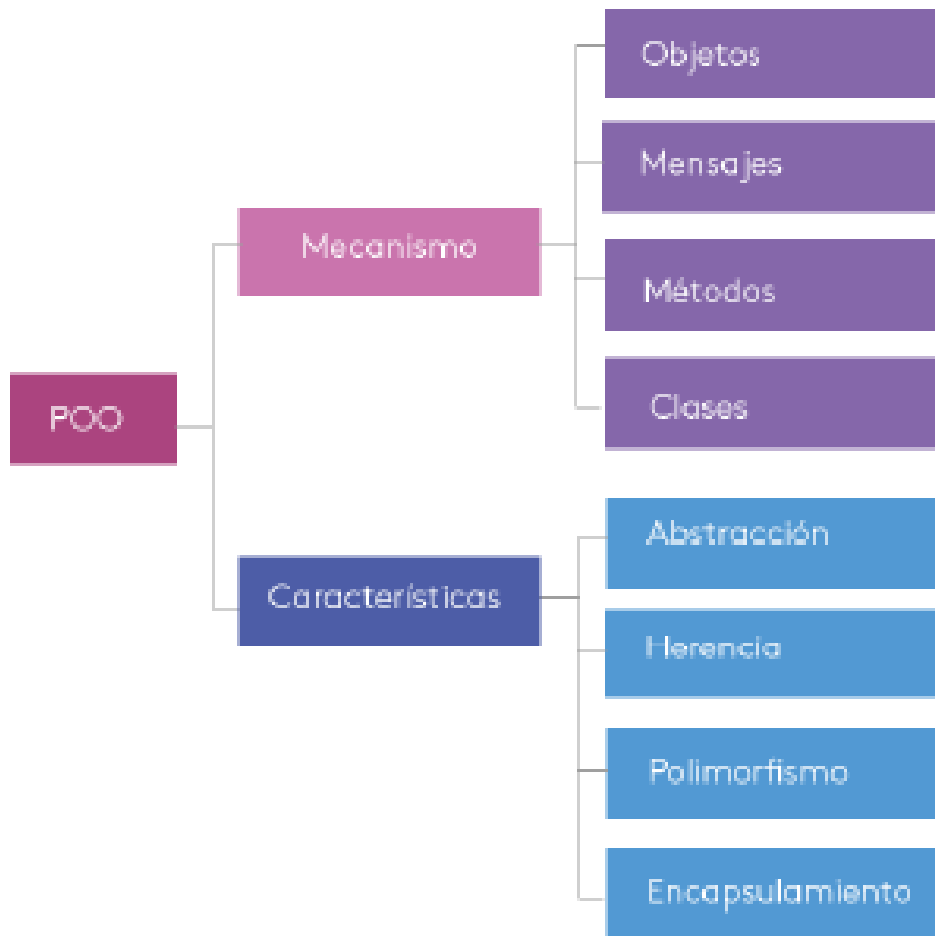
Cada vez que se construye un objeto a partir de una clase estamos creando lo que se llama una instancia de esa clase. La clase actúa como una plantilla, modelo o prototipo a partir del cual se obtienen instancias llamadas objetos.

Un ejemplo coloquial de aplicación de este concepto se puede realizar en la definición de características de un perro: para eso, se crea la clase llamada Perro, si bien es cierto que todos conocen a un perro, se puede agregar como dato específico el tipo raza. La clase general es Perro, pero se puede instanciar con la raza Pastor Alemán.

Si bien es cierto que en la POO hay mecanismos como objetos, métodos, mensajes y clases, hay también características como abstracción, herencia, encapsulamiento y polimorfismo. Esto lo podemos representar gráficamente así:



Imagen 2 Integrantes de la PO



Fuente: Elaboración propia

ATRIBUTOS DE UNA CLASE

Los atributos son fracciones de información que un objeto posee o conoce de sí mismo. Una clase puede tener cualquier número de atributos o no tener ninguno. Se declaran con un identificador y el tipo de dato correspondiente. Además, los atributos y tienen asociado un modificador que define su visibilidad.

La siguiente tabla especifica la visibilidad de los atributos de clase según su declaración:

Tabla 1 Visibilidad de los atributos de clase

Nombre	Descripción
Public - Pública	Se puede acceder al miembro de la clase desde cualquier lugar
Private- Privada	Sólo se puede acceder al miembro de la clase desde la propia clase.
Protected - Protegida	Sólo se puede acceder al miembro de la clase desde la propia clase o desde una clase que herede de ella.

Fuente: Elaboración propia

MÉTODOS

Los métodos representan acciones que pueden realizar los objetos, son procedimientos asociados a una clase y respetan una sintaxis en cada lenguaje específico. Se tiene acceso a ellos dependiendo de cómo están contruidos, es decir a través de su modificador de acceso: una palabra reservada que no se puede utilizar como variable porque pertenece al lenguaje como tal.

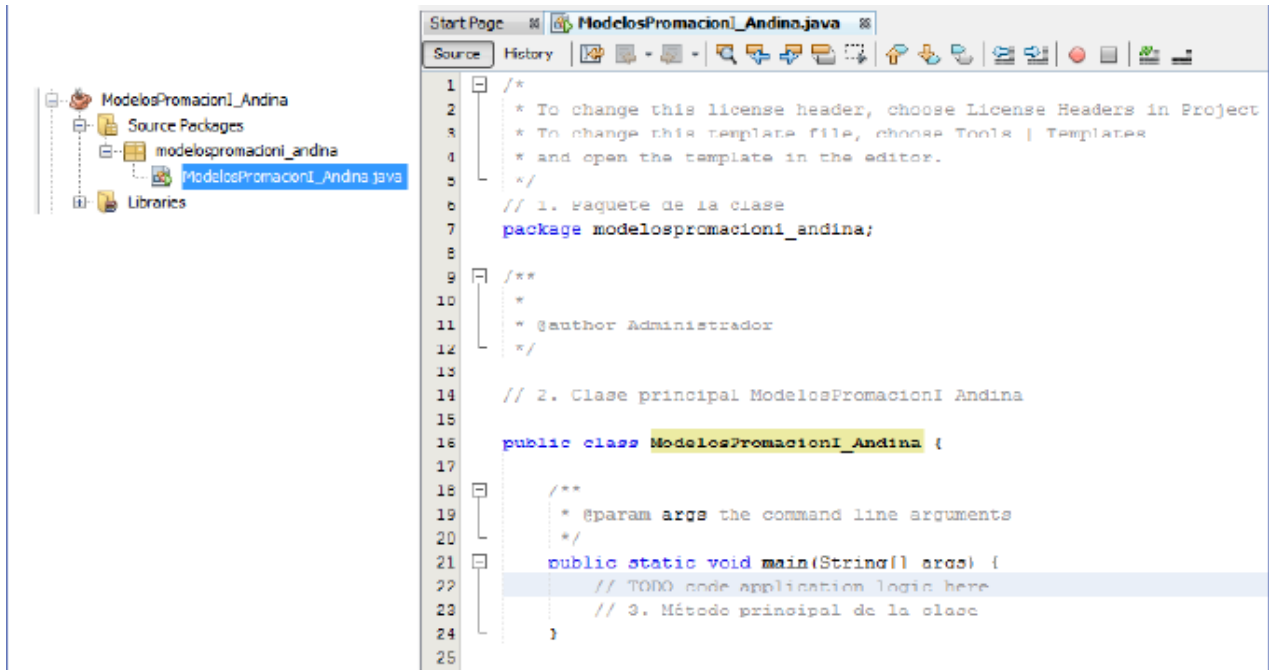
Un ejemplo de estas palabras reservadas son `private`, `public` y `protected` que relacionan ubicando al objeto que será utilizado por un determinado método.

Los métodos, igual que las variables, tienen un nombre que los identifica para poder ser llamados en la clase que los contiene. Los métodos pueden tener o no parámetros y retornos cuando son requeridos. En lenguaje Java con su IDE netbeans en la clase principal se encuentra el método `public static void main (String [] args)`.

La siguiente imagen con el método principal `main`, donde se visualiza la generación de un proyecto en Java netbeans para socializar su estructura y entorno. El proyecto parte de una clase principal, así como el método, esta

estructura se puede hacer desde cero por el programador o la misma plataforma le genera la estructura mínima como se observa en la figura 3.

Imagen 3 Clase Main principal en Java



```

1  /*
2  * To change this license header, choose License Headers in Project
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  // 1. Paquete de la clase
7  package modelospromacioni_andina;
8
9  /**
10 *
11 * @author Administrador
12 */
13
14 // 2. Clase principal ModelosPromacionI Andina
15
16 public class ModelosPromacionI_Andina {
17
18     /**
19     * @param args the command line arguments
20     */
21     public static void main(String[] args) {
22         // TODO code application logic here
23         // 3. Método principal de la clase
24     }
25

```

Fuente: Elaboración propia

CONSTRUCTORES:

Un constructor es una función, método o procedimiento de las clases, la cual es llamada automáticamente cuando se crea un objeto de esa clase. Por ser métodos, los constructores también aceptan parámetros. Cuando en una clase no especificamos ningún tipo de constructor, el compilador añade uno público por omisión sin parámetros, el cual no hace nada.

Algunas características importantes son:

- Posee el mismo nombre de la clase a la cual pertenece.
- No puede ser heredado.

- No retorna ningún valor, por lo cual no debe especificarse ningún tipo de dato.
- Es necesarios que se declare público, sólo en casos realmente extraordinarios será de otro tipo.

Imagen 4 Ejemplo de constructor

```
public class Datos
{
    //declaracion de atributos
    public Datos();//constructor sin parametros
    public Datos(int Edad)
    { //Constructor con parámetros
        EdadM=Edad;
    }
}
```

Fuente: <http://programandoenjava.over-blog.es>

DESTRUCTOR

El destructor se utiliza para destruir una instancia de una clase y liberar memoria. En Java no hay destructores, ya que la liberación de memoria es llevada a cabo por el Garbage Collector cuando las instancias de los objetos quedan des referenciadas.

Para ahondar más en el tema de Clases en POO debe realizar la lectura de las páginas 49 a la 88 del libro: Programación orientada a objetos con Java usando Bluej 6A. ED. (2017) de David J. Barnes y Micahel Kölling.

CONCLUSIONES Y RECOMENDACIONES

- Los objetos y las clases representan las bases de la programación orientada a objetos, este paradigma es bastante controlado y eficiente, el mercado actual busca su utilización en los nuevos desarrollos. Es por lo anterior que lenguajes de programación como Java tienen tanto auge en la actualidad.
- Java es un lenguaje de programación que dentro de su lógica utiliza en demasía las clases, de ahí la importancia de conocer y dominar este concepto.
- Las clases representan un concepto general de las cosas, mientras que los objetos representan instancias concretas de una clase.
- Una importante característica es la reutilización de código, por ejemplo, se puede definir una sola clase y de esa clase se pueden generar muchos objetos.
- Dentro del cuerpo de un constructor o método, una secuencia de instrucciones implementa la parte correspondiente a su comportamiento.

REFERENCIAS BIBLIOGRÁFICAS

- Aponte I, L. (16 de 11 de 2020). *Programando en Java*. Obtenido de <http://programandoenjava.over-blog.es/article-32829724.html>
- Barnes, D., & Kölling, M. (2017). *Programación oerientada a objetos con Java usando Bluej 6a. Ed.* Madrid: Pearson.
- Blasco, F. (2019). *Programación orientada a objetos en Java*. Bogotá: Ediciones de la U.
- Savitch, W. (2000). *Resolución de problemas con C++*. Madrid: Prentice Hall.
- Wu, T. (2001). *Introducción a la programación orientada a objetos con java*. Madrid: McGraw- Hill.





www.usanmarcos.ac.cr

San José, Costa Rica