	CENTRO COLOMBIANO DE ESTUDIOS PROFESIONALES APRENDER PARA CRECER	Fecha: 27-05-2023 Página 1 de 13
	DOCUMENTO INSTITUCIONAL	

Conceptos de la programación orientada a objetos

Sebastián Rodríguez Camilo
Dilan fierro facundo

Trabajo de consulta sobre Elementos de los conceptos de la programación
orientada a objetos (POO)


Morales León German Andrés

CECEP


FUNDACION CENTRO COLOMBIANO DE ESTUDIOS PROFESIONALES-
FCECEP, FACULTAD DE NTyC. TECNOLOGIA EN DESARROLLO DE
SISTEMAS INFORMATICO

Santiago de Cali

27/05/2023


	CENTRO COLOMBIANO DE ESTUDIOS PROFESIONALES APRENDER PARA CRECER	Fecha: 27-05-2023 Página 1 de 13
	DOCUMENTO INSTITUCIONAL	

1.1. INTRODUCCIÓN	4
CONCEPTO	5
¿Qué es la programación orientada a objetos?	5
Atributos:	5
Modificadores:	5
Tipo de variables:	5
Métodos:	6
ELEMENTOS	7
Encapsulamiento:	7
Private:	7
Public:	7
La herencia:	7
Clase:	7
¿QUE ES ITERACIÓN?	8
Iteración:	8
Tipos de Iteraciones (ciclos):	8
For:	8
while:	8
¿QUE SON LOS METODOS GETTERS Y SETTERS?	9
SETTERS (SET):	9
GETTERS (GET):	10
¿QUE ES UN FORMULARIO?	11
Tipos de formularios:	11
Formularios modales:	11
Formularios no modales:	11
Formularios padres:	11
Formularios hijos:	11
Elementos que contiene un formulario:	12
Etiqueta (JLabel):	12
Botón de dos posiciones (JToggleButton):	12
Botón de opción (JRadioButton):	12
Lista desplegable (JComboBox):	12

	CENTRO COLOMBIANO DE ESTUDIOS PROFESIONALES APRENDER PARA CRECER	Fecha: 27-05-2023 Página 1 de 13
	DOCUMENTO INSTITUCIONAL	

Campo de texto (JTextField):	12
Barra de Progreso (JProgressBox):	12
Cuadro de contraseña (JPasswordField):	12
Tabla (JTable):	12
MDI (JMenuBar):	12
Ventana (JFrame):	12
LAS IMPORTACIONES (IMPORT)	13
¿Qué es import Java util * en Java?:	13
Java Swing:	13
java.awt:	13
Algunas Importaciones:	14



	CENTRO COLOMBIANO DE ESTUDIOS PROFESIONALES APRENDER PARA CRECER	Fecha: 27-05-2023 Página 1 de 13
	DOCUMENTO INSTITUCIONAL	

1.1. INTRODUCCIÓN


Este trabajo consiste en explicar los conceptos a los elementos de la programación orientada a objetos (POO), precisamente a los elementos que sean venido incorporando en el programa de aprendizaje (desarrollo de sistemas informativos) de los estudiantes de la Fundación Centro Colombiano de Estudios Profesionales (Fcecep).

Con el objetivos de aprender y reconocer los conceptos de los distintos elementos que podemos encontrar a la hora de trabajar con la programación orientada a objetos.

Esto es importante porque: La POO es uno de los paradigmas de programación más utilizados en la actualidad y como estudiantes futuros programadores deben conocer los conceptos de cada elemento.

Usando fuentes de información e investigación podemos tomar los conceptos aprendidos para agregar a nuestra lengua de programadores y utilizar esta información para la realización de trabajos ya sea de lectura o escritura de códigos



	CENTRO COLOMBIANO DE ESTUDIOS PROFESIONALES APRENDER PARA CRECER	Fecha: 27-05-2023 Página 1 de 13
	DOCUMENTO INSTITUCIONAL	

CONCEPTO

¿Qué es la programación orientada a objetos?

La Programación Orientada a Objetos (POO) es un paradigma de programación que se basa en los objetos los cuales representa algo de la vida real y se utiliza para organizar y estructurar el código de manera Ordenada, Los objetos en la POO tienen dos características principales atributos y métodos.

Atributos:

Son las propiedades o características que posee un objeto. Pueden ser variables que almacenan datos como números, texto u otros objetos.

Sintaxis de un atributo: `[modificadores] tipo nombre;`

Modificadores:

Se utilizan para controlar el acceso, la visibilidad y el comportamiento de clases, métodos, atributos y constructores.

Ejemplo:

`private, protected, static, final`

Tipo de variables:

Numéricas:

`int`: Almacena números enteros

`float`: Almacena números con poca decimales

`double`: Almacena números de altas cantidades.


Variables de caracteres y cadenas:

`char`: Almacena un solo carácter (Unicode).

`String`: Almacena caracteres de texto.

Variables de booleanos:

`boolean`: Almacena un valor booleano (`true` o `false`).

	CENTRO COLOMBIANO DE ESTUDIOS PROFESIONALES APRENDER PARA CRECER	Fecha: 27-05-2023 Página 1 de 13
	DOCUMENTO INSTITUCIONAL	

Métodos:

Son las acciones o comportamientos que puede realizar un objeto. Los métodos son funciones que están asociadas al objeto y pueden modificar su estado o interactuar con otros objetos.

Sintaxis de métodos: [modificadores] tipo método (parametro);


Parámetros:

Un parámetro es una variable que se utiliza en la definición de un método para recibir valores desde el exterior cuando se llama al método.

Ejemplo:

```
Public int max (int x, int y) {
  if (x > y )
    Return x;
  Else
    Return y;
}
```



	CENTRO COLOMBIANO DE ESTUDIOS PROFESIONALES APRENDER PARA CRECER	Fecha: 27-05-2023 Página 1 de 13
	DOCUMENTO INSTITUCIONAL	

ELEMENTOS

Encapsulamiento: Es cuando limitamos el acceso o damos un acceso restringido de una propiedad a los elementos que necesita un miembro y no a ninguno más, Esto se logra con Private y public.


Private: Cuando se usa Private en un atributo o método se indica que solo es accesible desde dentro de la misma clase y no podrá ser vista por las otras.

Public: A diferencia de private el public permite que los atributos o métodos de una clase puede ser visto por otras clases y estos atributos podrán ser accesible para las demás clases de un proyecto.

La herencia: También conocida como subclase contiene atributos o métodos de la clase principal, La herencia tiene la capacidad para definir atributos y métodos nuevos para la subclase, que luego se aplican a los atributos y métodos heredados.

Clase: es un modelo que contiene que también define la estructura y el comportamiento de un objeto, Dentro de la clase se definen los atributos (propiedades) y los métodos (acciones) que tendrá el objeto de la clase.



	CENTRO COLOMBIANO DE ESTUDIOS PROFESIONALES APRENDER PARA CRECER	Fecha: 27-05-2023 Página 1 de 13
	DOCUMENTO INSTITUCIONAL	

¿QUE ES ITERACIÓN?

Iteración: La iteración se refiere a un proceso repetitivo que se realiza con el fin de ejecutar una serie de instrucciones varias veces. También se conoce como bucle o ciclo.

Tipos de Iteraciones (ciclos):

For: se utiliza sobre una secuencia de valores específica, como un rango de números o una colección de elementos (cuando sabemos cuántas veces se repetirá el ciclo "iteración").


Ejemplo de FOR:

```
for (int i = 0; i < 5; i++) {
    System.out.println("Iteración: " + i);
}
```

while: se utiliza para repetir un bloque de código mientras se cumpla una condición específica(hasta que el usuario tener de ejecutar el bloque de código).

Ejemplo de WHILE:

```
for (int i = 0; i < 5; i++) {
    System.out.println("Iteración: " + i);
}
```


	CENTRO COLOMBIANO DE ESTUDIOS PROFESIONALES APRENDER PARA CRECER	Fecha: 27-05-2023 Página 1 de 13
	DOCUMENTO INSTITUCIONAL	

¿QUE SON LOS METODOS GETTERS Y SETTERS?

SETTERS (SET):

Es utilizado para establecer o modificar (cambiar) el valor de un atributo o propiedad de un objeto. También se les conoce como métodos de configuración.

Ejemplos de SET:

setText(String texto): Este método se utiliza en componentes gráficos, como etiquetas o campos de texto, para establecer el texto que se mostrará en el componente.

setFont(Font fuente): Este método se utiliza para establecer la fuente (tipo y tamaño de letra) que se utilizará para mostrar el texto en un componente.

setForeground(Color color): Este método se utiliza para establecer el color del texto o de los elementos gráficos internos de un componente.

setIcon(Icon icono): Este método se utiliza para establecer un ícono o imagen que se mostrará en un componente, como un botón o una etiqueta.

setBorder(Border borde): Este método se utiliza para establecer el borde visual de un componente. Un borde puede ser una línea, una sombra, un borde redondeado, entre otros.

setBackground(Color color): Este método se utiliza para establecer el color de fondo de un componente.


setTitle(String titulo): Este método se utiliza en ventanas o cuadros de diálogo para establecer el título que aparecerá en la barra de título de la ventana.

setToolTipText(String texto): Este método se utiliza para establecer un texto descriptivo o informativo que se mostrará cuando el usuario coloque el cursor sobre un componente durante un tiempo determinado. El texto se muestra en un pequeño cuadro emergente conocido como "tooltip".

setLocationRelativeTo(Componente componente): Este método se utiliza en ventanas para establecer su posición relativa a otro componente. Si se pasa un componente como argumento, la ventana se posicionará en relación a ese componente.

setDefaultCloseOperation(int operacion): Este método se utiliza en ventanas para establecer la operación que se realizará cuando se cierre la ventana.

setVisible(boolean visible): Este método se utiliza en componentes gráficos, como ventanas o paneles, para establecer su visibilidad en la interfaz de usuario. Si se establece en true, el componente será visible; si se establece en false, será invisible.

	CENTRO COLOMBIANO DE ESTUDIOS PROFESIONALES APRENDER PARA CRECER	Fecha: 27-05-2023 Página 1 de 13
	DOCUMENTO INSTITUCIONAL	

GETTERS (GET):

Los GETTERS (GET) son métodos utilizados en la programación orientada a objetos para obtener o recuperar el valor de un atributo o propiedad de un objeto.

`getClass()`: Este método es heredado de la clase `Object` y se utiliza para obtener el objeto (`Class`) que representa la clase de un objeto en tiempo de ejecución.


`getResource(String ruta)`: Este método se utiliza para cargar recursos (archivos) desde el sistema de archivos o desde el classpath de la aplicación.

`getImage(URL url)`: Este método se utiliza en combinación con `getResource()` para cargar una imagen desde un recurso ubicado en una URL específica.

`getKeyCode()`: Este método se utiliza en el contexto de eventos de teclado en Java, como `KeyEvent`, para obtener el código numérico correspondiente a una tecla presionada.

`getText()`: Este método se utiliza en componentes de interfaz de usuario, como campos de texto o etiquetas, para obtener el texto contenido en el componente.



	CENTRO COLOMBIANO DE ESTUDIOS PROFESIONALES APRENDER PARA CRECER	Fecha: 27-05-2023 Página 1 de 13
	DOCUMENTO INSTITUCIONAL	

¿QUE ES UN FORMULARIO?

En la programación orientada a objetos los formularios son interfaces gráficas (Ventanas) de usuarios, En la que el usuario puede interactuar con el programa (Aplicación).

Tipos de formularios:


Formularios modales: estos se cierran o se ocultan para que el usuario pueda continuar trabajando con el programa.

Formularios no modales: Estos tipos de formularios permiten visualizar más de dos ventanas sin necesidad de cerrar la inicial.

Formularios padres: son aquellos que contienen formularios dentro de él a esto se le conocen como MDI.

Formularios hijos: estos están contenidos en otros formularios son conocidos Como SDI



	CENTRO COLOMBIANO DE ESTUDIOS PROFESIONALES APRENDER PARA CRECER	Fecha: 27-05-2023 Página 1 de 13
	DOCUMENTO INSTITUCIONAL	

Elementos que contiene un formulario:

Los formularios contiene campos de entrada de texto, botones, casillas de verificación y listas desplegables, etc.

Etiqueta (JLabel): se usa para visualizar un texto, imagen o ambas.

Botón de dos posiciones (JToggleButton): ES un botón con dos estados o dos opciones a elegir

Botón de opción (JRadioButton): Es un elemento puede ser seleccionado o deseleccionado usado.

Lista desplegable (JComboBox): Es un componente que cambia de botón a una lista desplegable.

Campo de texto (JTextField): Es un componente ligero que permite editar una línea individual de texto.

Barra de Progreso (JProgressBox): Este componente muestra a través de un porcentaje él informa al usuario el proceso de carga de un trabajo.


Cuadro de contraseña (JPasswordField): Es un componente que permite escribir sin Mostrar los caracteres ingresados.

Tabla (JTable): Es un componente utilizado para mostrar y editar datos ingresados.

MDI (JMenuBar): se utiliza comúnmente para crear y gestionar barras de menú en las ventanas principales de la aplicación.

Ventana (JFrame): Es una interfaces gráficas que será vista por el usuario.



	CENTRO COLOMBIANO DE ESTUDIOS PROFESIONALES APRENDER PARA CRECER	Fecha: 27-05-2023 Página 1 de 13
	DOCUMENTO INSTITUCIONAL	

LAS IMPORTACIONES (IMPORT)


¿Qué es import Java util * en Java?:

Las importaciones son que permiten utilizar y acceder a funcionalidades y componentes definidos en otros archivos o bibliotecas con el fin de ayudar a organizar y estructurar el código

Java Swing: es un conjunto de bibliotecas y componentes gráficos incluidos en la plataforma Java que se utilizan para desarrollar interfaces de usuario (UI).

java.awt: Es utilizado para crear ventanas, botones, campos de texto, etiquetas, paneles y otros componentes gráficos para diseñar la interfaz de usuario de tu aplicación.



	CENTRO COLOMBIANO DE ESTUDIOS PROFESIONALES APRENDER PARA CRECER	Fecha: 27-05-2023 Página 1 de 13
	DOCUMENTO INSTITUCIONAL	

Algunas Importaciones:

`java.awt.event.KeyEvent`: `KeyEvent`: es una clase que representa eventos de teclado en Java. Al importar `KeyEvent`, puedes utilizar sus constantes para identificar teclas

`import java.awt.Color`: Importa la clase `Color` se utiliza para representar y manipular colores en Java. Proporciona métodos para definir colores utilizando diferentes modelos de color, como RGB (Red, Green, Blue).

`import java.awt.Dimension`: Importa la clase `Dimensión` se utiliza para representar las dimensiones (ancho y alto) de los componentes gráficos en Java. Se utiliza para especificar el tamaño de un componente en píxeles.

`import java.awt.Font`: Importa la clase `Font` se utiliza para representar fuentes de texto en Java. Proporciona métodos para especificar el tipo de fuente, tamaño y estilo (negrita, cursiva, etc) de un texto.

`import java.awt.Image`: Importa la clase `Imagen` se utiliza para representar imágenes en Java. Proporciona métodos para cargar, manipular y mostrar imágenes en componentes gráficos.

`import java.awt.Toolkit`: Importa la clase `Toolkit` esta proporciona métodos para acceder a recursos y funcionalidades específicas del sistema, También se utiliza para obtener una instancia de la clase `Imagen` para representar el ícono de la aplicación.

`import java.awt.event.ActionEvent`: Importa la clase `ActionEvent` se utiliza para representar eventos de acción en Java. Los eventos de acción se generan cuando ocurre una interacción del usuario, como hacer clic en un botón.


`import java.awt.event.ActionListener`: Importa la interfaz `ActionListener` se usa para que un objeto que puede realizar un eventos de acción.

`import java.io.File`;: Importa la clase se utiliza para representar archivos y directorios en Java.

`import java.text.SimpleDateFormat`: Importa la clase `SimpleDateFormat` se utiliza para formatear y analizar fechas en Java. Permite convertir objetos `Date` en cadenas de texto formateadas y viceversa.

`Import java.util.Date`: Importa la clase `Date` se utiliza para representar fechas y horas en Java. Proporciona métodos para obtener y manipular componentes individuales de la fecha, como el año, el mes y el día.

`import javax.swing.ImageIcon`: Importa la clase `ImageIcon` que se utiliza para representar iconos en Java Swing

	CENTRO COLOMBIANO DE ESTUDIOS PROFESIONALES APRENDER PARA CRECER	Fecha: 27-05-2023 Página 1 de 13
	DOCUMENTO INSTITUCIONAL	

CONCLUSION

En conclusión en la programación orientada a objetos es importante reconocer los conceptos básicos sobre cada elemento, Como parte de la formación estudiantil y los requisitos que un programador debe cumplir, Esto con el fin de saber interpretar un código de forma en que podemos hacer partícipes ya sea a través de la lectura o escritura de un programa.

