

UNIVERSIDAD ANDINA DEL CUSCO

FACULTAD DE INGENIERIA Y ARQUITECTURA

CARRERA PROFESIONAL DE INGENIERIA DE SISTEMAS



TEMA:

"Fundamentos de programación orientada a objetos"

ASIGNATURA: ALGORITMICA Y LABORATORIO DE PROGRAMACION II

DOCENTE: ESPETIA HUAMANGA HUGO

ALUMNO: ZAPATA KIDA CESAR ALVARO

HUAMAN SUTTA ANTONY ESTIF

CUSCO –PERU

2019

Índice

Presentación:.....	3
Objetos:.....	4
Mensaje:	4
Ciclo de vida de un objeto:	4
Garbage Collector:.....	4
Destructores:	4
Métodos:	4
Clases:.....	4
Características de las POO:	5
Abstracción:.....	5
Encapsulamiento:	5
Cohesión y Acoplamiento:	5
Declaratividad y Expresividad:	5
La expresividad.....	5
La Declaratividad	5
Sobrecarga de métodos:.....	5
Composición.....	6
Herencia:.....	6
Ventajas	6
Desventajas	6
Composición.....	6
Polimorfismo:	6
Clase Abstracta:	7
Interfaces:	7
Mapa Mental:.....	7
Conclusión:	8
Bibliografía:.....	9

Presentación:

En el presente trabajo es el resumen del libro "Fundamentos de programación orientada a objetos" por Nicolás Orchard, Nicolás Bonanno, Santiago Rojo y Ramiro Berruezo” tiene como objetivo como principal analizar e interpretar la información con objeto de que sirva a para realizar la programación orientada a objetos (POO).

Con este libro obtuvimos conocimientos fundamentales que nos introducen ala Programacion orientadas a objetos.

Objetos:

La definición más básica dice: "Es un ente computacional que puede contener datos y comportamientos"

Mensaje:

Todos los objetos de los cuales hablamos tienen comportamientos y se relacionan con otros objetos

Ciclo de vida de un objeto:

Garbage Collector: es un mecanismo que se encarga de borrar de la memoria las referencias a objetos y entidades que ya no se usan más, de manera que se pueda maximizar el uso del espacio en memoria.

Destructores: son métodos que se definen para cada objeto y cuyo principal objetivo es liberar los recursos que fueron adquiridos por el objeto a lo largo de su ciclo de vida y romper vínculos con otras entidades que puedan tener referencias a él.

Métodos:

Un método es la sección de código que se ejecuta al enviar un mensaje

La firma de un objeto se define con tres componentes:

- El nombre del método.
- Los parámetros que recibe el método.
- Lo que devuelve el método (que puede ser nada u otro objeto).

Clases:

Una clase es la descripción de un conjunto de objetos similares; consta de métodos y de datos que resumen las características comunes de dicho conjunto.

Características de las POO:

Abstracción:

La abstracción es la propiedad de los objetos que consiste en tener en cuenta sólo los aspectos en el proceso de abstracción permite seleccionar las características relevantes dentro de un conjunto e identificar comportamientos comunes para definir nuevos tipos de entidades en el mundo real. La abstracción es clave en el proceso de análisis y diseño orientado a objetos, ya que mediante ella podemos llegar a armar un conjunto de clases que permitan modelar la realidad o el problema que se quiere atacar.

Encapsulamiento:

Encapsulamiento es la capacidad de diferenciar qué partes de un objeto son parte de la interfaz y cuales permanecerán inaccesibles por el usuario, significa reunir todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Esto permite aumentar la cohesión de los componentes del sistema.

Cohesión y Acoplamiento:

La cohesión mide la relación entre el concepto que deseamos modelar y las responsabilidades del componente que lo representan.

El acoplamiento mide qué tan relacionados están los componentes del sistema entre sí y cómo esa dependencia provoca un diseño.

Declaratividad y Expresividad:

La expresividad

Tiene que ver con qué tan claro es el código.

La Declaratividad

Ayuda a que el código sea más expresivo, porque no se mezcla el algoritmo con lo que quieres que haga más a alto nivel.

Sobrecarga de métodos:

La sobrecarga es la capacidad de un lenguaje de programación, que permite nombrar con el mismo identificador diferentes variables u operaciones. La sobrecarga de métodos se refiere a la posibilidad de tener dos o más métodos con el mismo nombre pero diferente firma.

Composición

La composición se refiere a la combinación de objetos simples para hacer objetos más complejos.

Herencia:

Es el mecanismo por el cual un objeto se basa en otro objeto o clase, extendiendo la implementación para reutilizar el comportamiento.

Ventajas

- Se define en tiempo de compilación y no tiempo de ejecución, es decir que conocemos con seguridad que se va a ejecutar.
- Es más sencillo de modificar la implementación que está siendo reutilizada.

Desventajas

- Como dijimos antes, se definen en tiempo de compilación, por lo que no se puede cambiar el comportamiento de un objeto en tiempo de ejecución, lo cual puede ser un requisito de nuestro diseño.
- Se rompe el concepto de encapsulación al exponer los detalles de la implementación en la clase padre.

Composición

Conocida como “reutilización de caja negra”, a diferencia del caso anterior nosotros no conocemos los detalles de la implementación, ya que la misma se encuentra encapsulada en el objeto al cual estamos invocando.

Polimorfismo:

Es la capacidad que tiene un objeto de poder tratar indistintamente a otros que sean potencialmente distintos, es decir, es la capacidad que tienen distintos objetos de entender un mismo mensaje.

Clase Abstracta:

Las clases abstractas son aquellas que por sí mismas no se pueden identificar con algo 'concreto' (no existen como tal en el mundo real), pero sí poseen determinadas características que son comunes en otras clases que pueden ser creadas a partir de ellas.

Interfaces:

Una interfaz define un tipo y es una colección de métodos abstractos.

Mapa Mental:

A partir del siguiente concepto desarrollamos un mapa mental para poder verlo se debe de dirigir al siguiente link:

- <https://prezi.com/p/yzstpoka6p8e/poo/>

Conclusión:

Gracias este libro obtuvimos nuevos conocimientos acerca de la Programacion orientada a objetos la cual tiene características como la herencia, polimorfismo, abstracción y el encapsulamiento.

Bibliografía:

- Nicolás Orchard, Nicolás Bonanno, Santiago Rojo y Ramiro Berruezo. (2014). Fundamentos de Programación Orientada Objetos. 23/08/2019, de Graion Sitio web
:file:///C:/Users/akida/Downloads/Fundamentos+de+programaci%C3%B3n+orientada+a+objetos+-+Nicol%C3%A1s+Orchow,+Nicol%C3%A1s+Bonanno,+Santiago+Rojo,+Ramiro+Berruezo.pdf
- Informaticapc. (2012). introducción a la Programación Orientada a Objetos (POO) y UML Clases abstractas. 23/08/2018, de informática Sitio web:
<https://informaticapc.com/poo/clases-abstractas.php>