******

***Universidad Autónoma de Santo Domingo***

*Primada de América Fundada 28 octubre 1538*

***UASD***

***FACULTAD DE CIENCIAS***

***ESCUELA DE INFORMÁTICA***

***LABORATORIO DE PROGRAMACIÓN II***

***INF-515***

***Sec. 0I***

***Practica de laboratorio Final CRUD con base de datos MySQL en java***

***SUSTENTANTE***

***Alexander Solano de la Cruz AB-******5972***

***PROFESOR***

***Lic. Silverio Del Orbe Abad***

*REPÚBLICA DOMINICANA*

*Mayo, 2024*

LAB Final de Semestre

*Autor: Alexander Solano de la Cruz*

Matricula: AB5972

*Concepto general aprendido*:

El concepto de CRUD se refiere a las operaciones básicas utilizadas en el desarrollo de software y la gestión de bases de datos relacionales. CRUD es un acrónimo que representa las siguientes operaciones:

Crear (Create): Esta operación implica la creación o inserción de nuevos registros en una base de datos.

Leer (Read): Esta operación se utiliza para consultar o leer registros existentes en una base de datos.

Actualizar (Update): Esta operación implica la modificación de registros existentes en una base de datos.

Eliminar (Delete): Esta operación se utiliza para eliminar registros existentes de una base de datos.

Declaración de clases abstractas que es una clase común la cual posee atributos, métodos, constructores y por lo menos un método abstracto. Una clase abstracta no puede ser instanciada, solo heredada. Una clase abstracta puede tener métodos abstractos, que son métodos que no tienen una implementación y deben ser implementados por las clases que heredan de la clase abstracta.

declaración de variables de instancias, declaración de un método override - sobreescritura, declaración y uso de la herencia[clases padres, clases hijas]

*Utilidad del Caso práctico*:

La importancia del CRUD radica en su capacidad para gestionar de manera efectiva la información almacenada en una base de datos. Estas operaciones son esenciales para garantizar la integridad y consistencia de los datos, así como para permitir a los usuarios interactuar y manipular la información de manera segura y eficiente.

Uso de clases abstractas que permiten al programador decidir qué clases van a poder ser instanciables (se van a poder crear objetos de ellas) y cuáles no (es decir, van a servir sólo para hacer de soporte para programar nuevas clases por herencia).

La herencia permite que se puedan definir nuevas clases basadas de unas ya existentes a fin de reutilizar el código, generando así una jerarquía de clases dentro de una aplicación. Si una clase deriva de otra, esta hereda sus atributos y métodos y puede añadir nuevos atributos, métodos o redefinir los heredados.

La sobreescritura de métodos override nos permite a los personalizar el comportamiento de los métodos en una clase derivada (subclase) para adaptarlos a las necesidades específicas de esa clase.

Los métodos con sobreescritura son: getArea(), getPerimetro(), getCantidadDeLados(), getTipos()

Descripción del componente: instance fields (cantidad) - en este programa utilizamos – 3 campos de instancias para la figura paralelogramo , los cuales son private, para poder ser heredados por las clases hijas:

private double lado1;

private double lado2;

private double lado3;

3 campos de instancias para la figura trapecio , los cuales son private, para poder ser heredados por las clases hijas:

private double baseMayor;

private double baseMenor;

private double altura;

Los metodos constructores son:

public Paralelogramo(double lado1, double lado2, double lado3)

Los métodos setters y getters son: setLado, getLado,

Las operaciones son los metodos:

public abstract double getArea();

public abstract double getPerimetro();

public abstract int getCantLados();

public abstract String getTipo();

***Codigo Fuente***

/\*

Laboratorio de Programacion 2

LAB Final

Autor: Alexander Solano de la Cruz

Matricula: AB5972

Prof. Lic. Silverio DelOrbe Abad

Fecha: 02/05/2024

\*/