



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

José Antonio Ayala Barbosa

Profesor:

programación orientada a objetos

Asignatura:

Gpo 1 22-2

Grupo:

04

No de Práctica(s):

Rosillo Montijo Emmanuel Alonso

Integrante(s):

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada:

03

Semestre:

17/02/2022

Fecha de entrega:

Observaciones:

CALIFICACIÓN: _____

Objetivo:

Aplicar los conceptos básicos de la programación orientada a objetos en un lenguaje de programación.

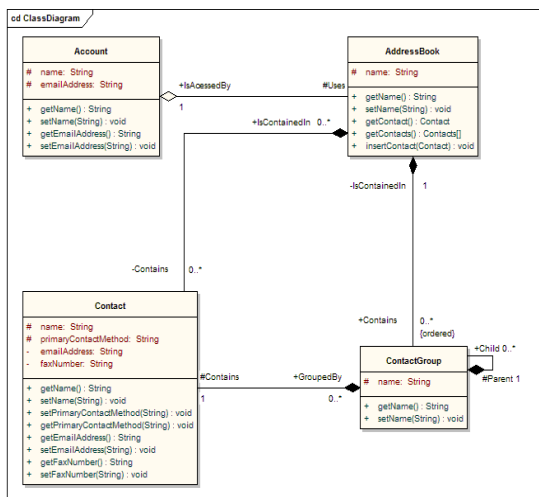
Previo:

Diagramas de Clases con al menos 3 atributos y 3 métodos

Introducción:

Para poder llevar acabo esta práctica se necesita conocer cómo funcionan los diagramas de clases y sus componentes, y si revisamos un poco, podemos observar que los diagramas de clases son uno de los tipos de diagramas más útiles en UML, ya que trazan claramente la estructura de un sistema concreto al modelar sus clases, atributos, operaciones y relaciones entre objetos.

Los ejemplos gráficos son los siguientes



Que consta de tres partes:

- Sección superior: Contiene el nombre de la clase. Esta sección siempre es necesaria, ya sea que estés hablando del clasificador o de un objeto.
- Sección central: Contiene los atributos de la clase. ...
- Sección inferior: Incluye operaciones de clases (métodos).

Para poder usarlos y comprender su funcionamiento se llevan acabo las actividades de esta práctica ya que con el diagrama UML en esta práctica nos sirve para representar el objeto en el diagrama

Actividades:

Se lleva a cabo la creación de 5 clases distintas, cual el nombre de las clases son la sección superior de esta forma.

```
public class Alumno {
```

```
public class Perro {
```

Las clases son la de **perro**, **alumno**, **profesor**, **triangulo**, **persona**

Cada una de las clases recibe el nombre del objeto que se quiere crear.

Una vez generada la clase se crean sus atributos con su respectivo tipo de dato, en UML es la segunda sección

De esta forma:

```
public class Perro {  
    String raza, color, nombre;  
    Float altura, peso;  
}
```

```
public class Profesor {  
    String nombre, materia, grupo, salon;  
    int noDeAlumnos;  
}
```

Para indicar que los atributos le pertenecen a la clase se crea el método constructor para los atributos:

```
public Profesor() {  
    this.nombre = nombre;  
    this.materia = materia;  
    this.grupo = grupo;  
    this.noDeAlumnos = noDeAlumnos;  
}
```

De esta forma se realiza un constructor.

En la tercera sección del UML encontramos el código que lleva los métodos de la clase, serán las partes

podrán manipular los atributos para realizar ciertos procesos:

```
public Triangulo() {  
    this.base = base;  
    this.altura = altura;  
}  
public void perimetro() {  
    int perimetro = base * 3;  
    System.out.println("perimetro total: " + perimetro);  
}  
public void area(){  
    int areaTotal = (base * altura)/2;  
    System.out.println("area total: " + areaTotal);  
}
```

Una vez creadas las clases, podemos instanciarlas creando un nuevo objeto, el cual va a contener los atributos de la clase y podrá utilizar los métodos de las mismas.

De esta forma

```
System.out.println("#####Objeto Alumno#####");  
Alumno lopez = new Alumno();  
lopez.nombre = "lopez";  
String nombreLopez = lopez.nombre;  
System.out.println(nombreLopez);
```

El objeto adquiere el valor del atributo que se le asigne de esta forma:

```
lopez.nombre = "lopez";
```

Se pueden utilizar los métodos así:

```
System.out.println("#####Objeto Alumno#####");  
Alumno lopez = new Alumno();  
lopez.nombre = "lopez";  
String nombreLopez = lopez.nombre;  
System.out.println(nombreLopez);  
  
System.out.println("#####Objeto Profesor#####");  
Profesor jirafales = new Profesor();  
jirafales.nombre = "jirafales";  
jirafales.grupo = "a1";  
jirafales.noDeAlumnos = 12;  
Profesor.calificacionFinal(nombreLopez);  
}
```

Para esta práctica cree los cinco objetos, y devuelven lo siguiente:

```
#####Objeto Triangulo#####
perimetro total: 36
area total: 36
#####Objeto Persona#####
boxeador: Pepe chuchin
De peso: 72.5
Y de altura: 1.76
Peleara por el cinturón de oro
Este es su índice de masa corporal de acuerdo a sus datos: 23.405218
Tiene permitido comer una dieta omnívora
por lo que comerá una dieta equitativa en todos sus aspectos
#####Objeto Perro#####
el perro pipo ladra muy bajito porque es un chihuahua
#####Objeto Alumno#####
Lopez
#####Objeto Profesor#####
Al calificar a Lopez se obtuvieron los sig. resultados:
parcial 1: 10
parcial 2: 10
parcial 3: 10
Por lo que obtuvo un promedio de: 10.0
después de una larga jornada de estudio el alumno: Lopez promedio

P00 10.0
edai 2 9
cyg 8

Process finished with exit code 0
```

En la clase alumno y profesor, se ligaron por así decirlo, de la siguiente forma:

```
public void enseñar(String nombreLopez) {
    System.out.println("el profesor: " + nombre + " está de enseñando al grupo " + grupo
        + " \nactual tiene un número de alumnos: " + noDeAlumnos);
    System.out.println("En este grupo se encuentra " + nombreLopez +
        "Está inscrito para tomar el curso");
}

public static void calificacionFinal(String nombreLopez) {
    int a, b, c;
    a = 10;
    b = 10;
    c = 10;
    System.out.println("Al calificar a " + nombreLopez +
        " se obtuvieron los sig. resultados: ");
    System.out.println("parcial 1: " + a + "\nparcial 2: " + b + "\nparcial 3: " + c);
    float cal = Profesor.promediar(a, b, c);
    System.out.println("Por lo que obtuvo un promedio de: " + cal);
    Alumno.promedio(cal);
}

public static float promediar(int a, int b, int c) {
    float calculo = (a + b + c) / 3;
    return (calculo);
}
```

Se llama a nombre de López

Se imprime sus parciales dentro del método del profesor se calcula el promedio y se retorna como atributo a López.

de esta forma:

```
public static float promedio(float cal) {
    int b = 9;
    int c = 8;
    System.out.println("después de una larga jornada de estudio el alumno: " + nombre + " promedio\n");
    System.out.println("P00 " + cal);
    System.out.println("edai 2 " + b);
    System.out.println("cyg " + c);
    float calificacionesP00 = (cal + b + c) / 3;
    Alumno.calificaciones = calificacionesP00;
    return (calificacionesP00);
}
```

Y ahora López tiene el promedio como atributo que le asignó su maestro

Conclusión:

Los objetos y las clases nos ayudan a abstraer la realidad con mayor facilidad al código, para que de esta forma podamos simplificar procesos y generar nuevos para crear programas más complejos