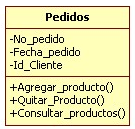
# TALLER PROGRAMACION ORIENTADA A OBJETOS

1. Para que un objeto pertenezca a una clase debe ser declarado e instanciado. Consulte que es el método constructor en una clase y cuantos métodos constructores puede tener una clase.

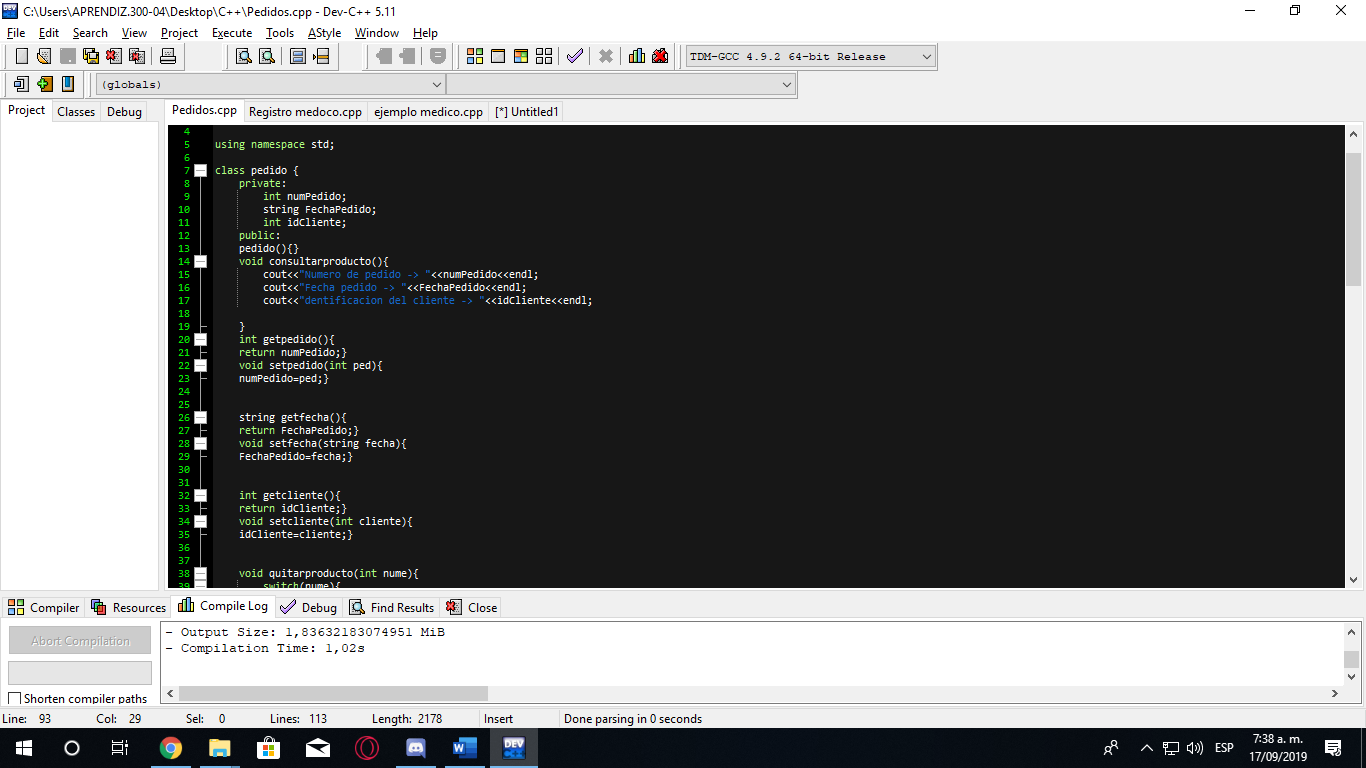
## Encapsulación

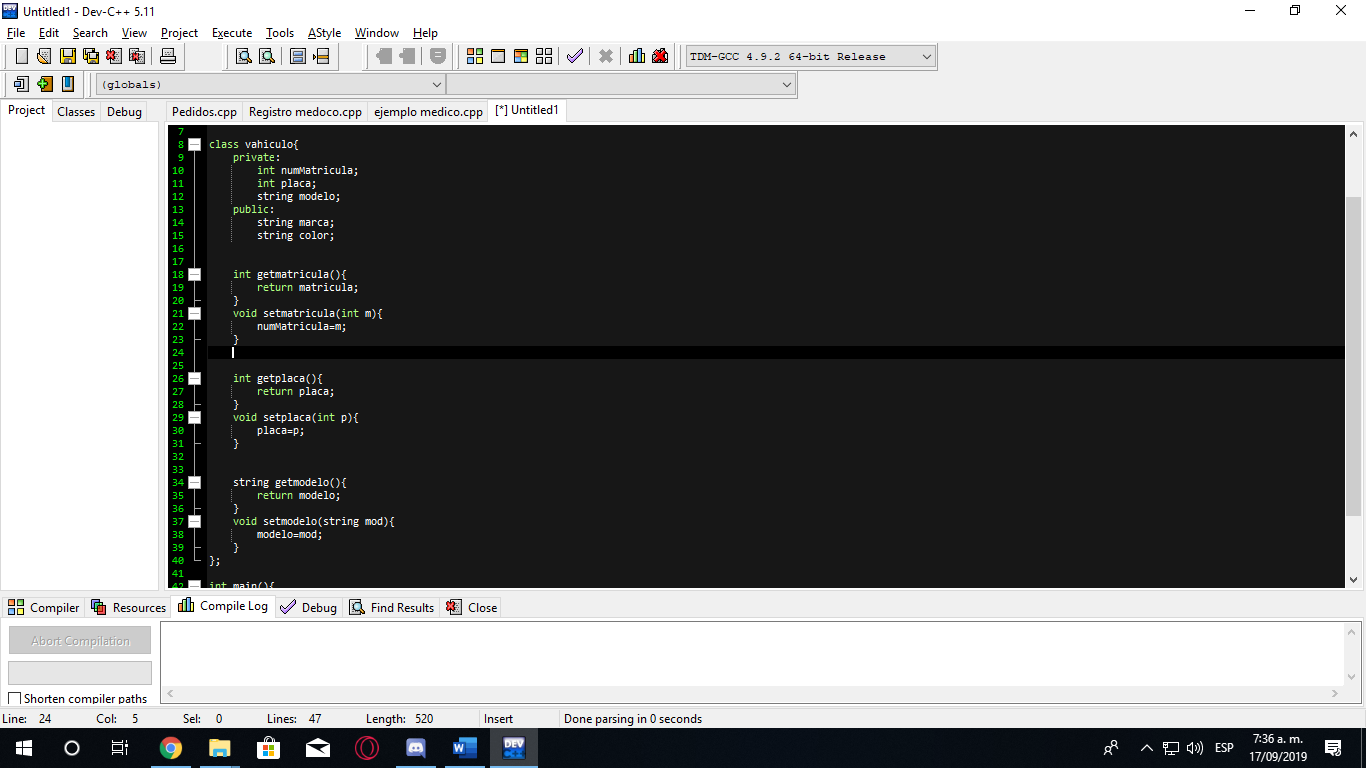
1. Los miembros de una clase, son el conjunto de atributos y métodos de esa clase. Para acceder a los miembros de una clase existe el concepto “visibilidad”, consulte sobre los siguientes tipos de visibilidad:
   * Publico
   * Privado
   * Protegido
2. Identifique el tipo de visibilidad del atributo para las siguientes situaciones:
   * El nombre del cliente lo pueden consultar desde cualquier ámbito.
   * El número de cuenta solo puede ser modificado dentro de la clase.
   * El saldo puede ser visto desde las subclases.
3. ¿Qué es el encapsulamiento en la programación Orientada a Objetos?

¿Cómo se aplica para las clases? ¿Cómo se aplica para los atributos de la clase?

1. Cuál es el código en C++ para declarar las siguientes clases, adicionar los métodos que permitirían el acceso a los atributos privados de la clase (get/set), para los otros métodos incluir un mensaje de texto indicando la utilidad del método:
2.  b)

a)



b)

1. Se requiere construir una clase llamada “cuentaCorriente”, en ella se debe tener la información del número de cuenta, saldo actual, sobregiro autorizado, clave secreta y la sucursal del banco de donde es la cuenta. Se debe contar con métodos para *consignarDinero*, para *retirarCajero* y para *pagarCheque*. El primer método (*consignarDinero*)debe permitir incrementar el saldo en un valor especifico, el método *retirarCajero* debe disminuir el saldo en un valor especifico, pero solo si la clave coincide y el método *pagarCheque* debe disminuir el saldo en un valor dado. ¿Cuál es el código en lenguaje C++ para construir esta clase con la funcionalidad requerida?

#include<iostream>

#include<stdlib.h>

using namespace std;

class cuentacorriente{

private:

int numCuenta;

double saldo;

double giro;

int clave;

public:

cuentacorriente(){}

string banco;

int getcuenta(){

return numCuenta;

}

void setcuenta(int c){

numCuenta=c;

}

double getsaldo(){

return saldo;

}

void setsaldo(double s){

saldo=s;

}

double getgiro(){

return giro;

}

void setgiro(double g){

giro=g;

}

int getcalve(){

return clave;

}

void setclave(int cla){

clave=cla;

}

void consignardinero(double dinero1){

saldo=saldo+dinero1;

cout<<"este es su saldo total -> "<<saldo<<endl;

}

void retirardinero(int cl){

if (cl=clave){

cout<<"valor a retirar"<<endl;

double retiro;

cin>>retiro;

saldo=saldo-retiro;

cout<<"este es su saldo total -> "<<saldo<<endl;

}

else {

cout<<"la clave es incorrecta"<<endl;

}

}

void pagarcheque(double cheque){

saldo=saldo+cheque;

cout<<"este es su saldo total -> "<<saldo<<endl;

}

};

int main(){

cuentacorriente invocar;

int c,num,cla, cl;

double s,g, dinero1, cheque;

string banco;

cout<<"-------------BANCO--------------"<<endl;

cout<<"-----------REGISTRO--------------"<<endl;

cout<<"1. ingrese su numero de cuenta "<<endl;

cin>>c;

invocar.setcuenta(c);

cout<<"2. cuato saldo tiene "<<endl;

cin>>s;

invocar.setsaldo(s);

cout<<"3. sobre giro autorizado "<<endl;

cin>>g;

invocar.setgiro(g);

cout<<"4. ingrese su clave de solo numeros"<<endl;

cin>>cla;

invocar.setclave(cla);

cout<<"5. nombre del banco"<<endl;

cin>>banco;

cout<<"gracias por su informacion "<<endl;

system("cls");

system("pause");

while(num !=99){

cout<<"--------Banco:"<<banco<<"---------"<<endl;

cout<<"Escoja un numero para continuar "<<endl;

cout<<"1. consignacion del dinero "<<endl;

cout<<"2. Retirar del cajero "<<endl;

cout<<"3. Pagar cheque "<<endl;

cin>>num;

switch(num){

case 1:{

cout<<"cuanto desea consignar "<<endl;

cin>>dinero1;

invocar.consignardinero(dinero1);

break;

}

case 2:{

cout<<"cual es su clave" <<endl;

cin>>cl;

invocar.retirardinero(cl);

break;

}

case 3:{

cout<<"de cuento sera el cheque "<<endl;

cin>>cheque;

invocar.pagarcheque(cheque);

break;

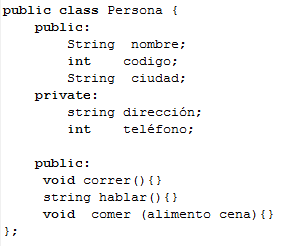
}

system("cls");

}

}

}



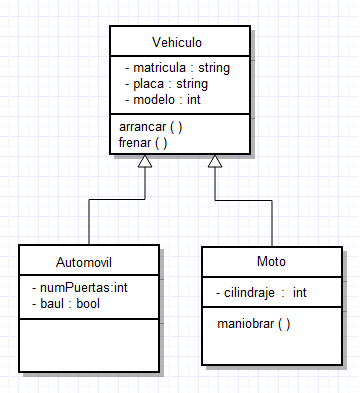
1. Para el siguiente segmento de código, represente gráficamente la clase que se implementa.

|  |
| --- |
| PERSONA |
| + nombre  + código  + ciudad  - dirección  - teléfono |
| Correr ()  Hablar ()  Comer () |

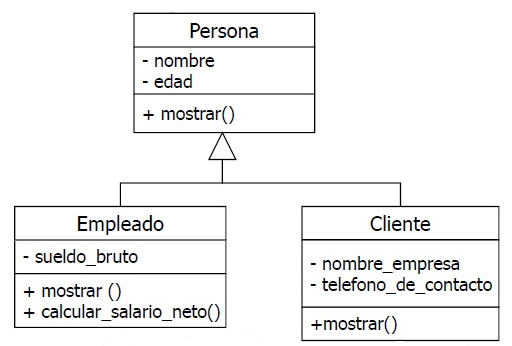
1. Construir una clase que contenga un arreglo con 10 posiciones. El constructor debe inicializar el arreglo con los números del 1 al 10. Se debe incluir un método que permita modificar una posición del arreglo con un nuevo valor.

## Herencia

1. ¿Qué es la generalización en las clases? ¿Qué es herencia? ¿Qué es una clase base? ¿Qué es una clase derivada? ¿Cómo se indica en lenguaje C++ que una clase hereda de otra?
2. Para los siguientes diagramas construya las clases en lenguaje C++ que representen el modelo y sus relaciones.



a)

b) 

1. Se requiere construir un programa en C++ con las clases necesarias para dar respuesta a la siguiente situación:

Una empresa editorial requiere manejar la información de sus **publicaciones**, se sabe que todas las publicaciones tienen un título, autor, edición y código ISBN. Para todas las publicaciones se debe tener un método para el lanzamiento que muestre los detalles del evento de lanzamiento.

Existen tres tipos de publicaciones, cuando son **libros** se debe almacenar un resumen y la edición. Para los **cuentos**, se requiere almacenar el nombre del ilustrador y el género del cuento. Cuando son **revistas** es necesario identificar el volumen, número y año. Para las clases derivadas se requiere implementar un método para consultar los datos y otro para asignar los valores. (Se deben incluir algunos atributos como privados).

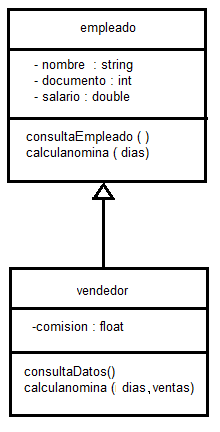
En el bloque principal del programa se debe incluir un menú para seleccionar el tipo de publicación y las operaciones que se pueden realizar con ellas.

## Polimorfismo

1. ¿Qué es polimorfismo en programación orientada a objetos? Consulte dos ejemplos.
2. Represente el siguiente diagrama de clases en lenguaje C++, tenga en cuenta que se debe generar la herencia y el método de la clase derivada debe modificar la funcionalidad respecto la funcionalidad de la clase base. En la clase base un empleado tiene nombre documento y salario, como métodos una ***consultaEmpleado*** que presenta los datos de un empleado y el método ***calculanomina*** permite calcular su salario mensual en función del número de días que trabajó.

La clase derivada: vendedor incluye el porcentaje de comisión que recibe. Se debe implementar el método ***consultaDatos*** que debe mostrar sus datos completos (nombre, documento, salario) que están en la clase base y el porcentaje de comisión que está en su clase).

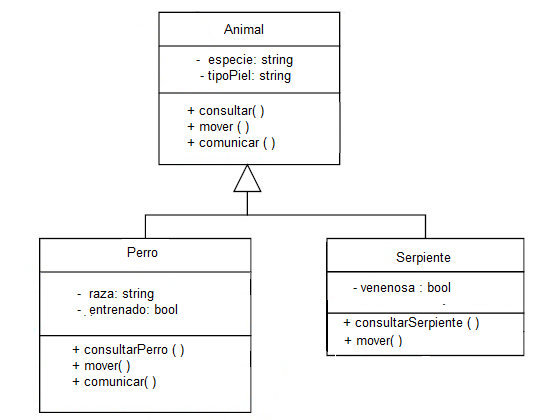
El método ***calculanomina*** del vendedor debe modificarse ya que es necesario recibir las ventas sobre las cuales calcular la comisión que se suma al salario que le corresponde por el número de días trabajado.



1. Dado el siguiente diagrama, implemente el código para construir la estructura de clases: Tener en cuenta que el método consultar en la clase animal muestra los datos básicos del animal, pero consultarPerro( ) debe mostrar los datos heredados así como los propios. Igual ocurrirá con la función consultarSerpiente ( ).

El método mover de la clase Animal debe escribir: “el animal puede moverse”. Para implementar el concepto de polimorfismo se debe construir el método mover del perro recibiendo un numero entero, para que se escriba el mensaje “el perro se desplaza con sus **N** extremidades”, donde N es el numero que llega al método. El método comunicar del perro debe recibir un string con el sonido del ladrido. Para la clase serpiente el método mover debe recibir el texto “vientre”.

Construir el código en C++;



1. Proponer una estructura de herencia relacionada con el proyecto de formación, debe incluir mínimo dos clases e implementar algunos métodos, al menos 3. Se debe presentar el texto que describe la situación, las clases asociadas y el código implementado en C++.