Instituto Politécnico Nacional

Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas

“UPIITA”

Ingeniería Mecatrónica

Asignatura: Programación Avanzada

“Tema 5: Abstracción”

Profesora: Maza Casas Lamberto

Grupo: 2MV3

Alumnos:

Alonso Bernal Alejandro

López Piedracruz Marcos Antonio

Luviano Murakawa Tsuioshi Alberto

**Introducción**

En el siguiente reporte de práctica se tocará un tema importante para la programación orientada objetos que es la abstracción.

La abstracción se define como las características específicas de un objeto, aquellas que lo distingue de los demás tipos de objetos y que logran definir límites conceptuales respecto a quién está haciendo dicha abstracción del objeto.

En abstracción existe una barrera de abstracción la cual se enfoca la visión externa de un objeto, separa el comportamiento específico de un objeto, la cuál se consigue aplicando el principio de mínimo compromiso.

El mínimo compromiso se refiere al proceso por el cual la interfaz de un objeto muestra su comportamiento específico.

Una interfaz de objeto permite crear código con el cuál se especifica que métodos serán implementados por una clase sin necesidad de definir que harán estos elementos, dichos métodos pueden ser públicos.

Existen varios tipos de abstracciones:

**Abstracción de entidades:** Es un objeto que representa un modelo útil de una entidad que se desea.

**Abstracción de acciones:** Un objeto que representa un conjunto de operaciones y todas desempeñan funciones del mismo tipo.

**Abstracción de Máquinas virtuales:** Un objeto que agrupa operaciones, todas ellas virtuales, utilizadas por algún nivel superior de control u operaciones (entre ellos podríamos hablar de un circuito)

**Desarrollo**

/\* Programa principal \*/

#include <iostream>

#include <string>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

using namespace std;

#include "claseEjAbstr.h"

int main(int argc, char\*\* argv) {

#ifdef EJEMPLO

char letter1,letter2; //declaración de dos variables tipo carácter

string lastName;

//Enter letters and print message

cout<<"Teclea dos iniciales y un nombre: ";

cin>>letter1>>letter2>>lastName; //Se introducen las dos variables

cout<<"Hola "<<letter1<<"."<<letter2<<"."

<<lastName<<"!"<<endl; // se imprimen los dos caracteres que se introdujeron

cout<<"Esperamos que disfrute estudiando C++."<<endl;

#endif

ClaseEjAbstr \*CEA=new ClaseEjAbstr(20); // se crea un apuntador a ClaseEjAbstr

ClaseEjAbstr obj(30);

cout<<"Atributo cantidad="<<CEA->get\_cantidad()<<"\n";

cout<<"Atributo cantidad="<<obj.get\_cantidad()<<"\n";

return 0;

}

/\* header \*/

/\*\*

\* claseEjAbstr.h

\*/

class ClaseEjAbstr{ //se crea la clase ClaseEjAbstr

//int cantidad;

int cantidadArr[2];

public:

int get\_cantidad();

ClaseEjAbstr(int);

};//end class ClaseEjAbstr

/\*claseejabstr \*/

/\*\*

\* claseEjAbstr.cpp

\*/

#include "claseEjAbstr.h"

ClaseEjAbstr::ClaseEjAbstr(int cantidad){

//this->cantidad=cantidad;

cantidadArr[0]=cantidad;

}

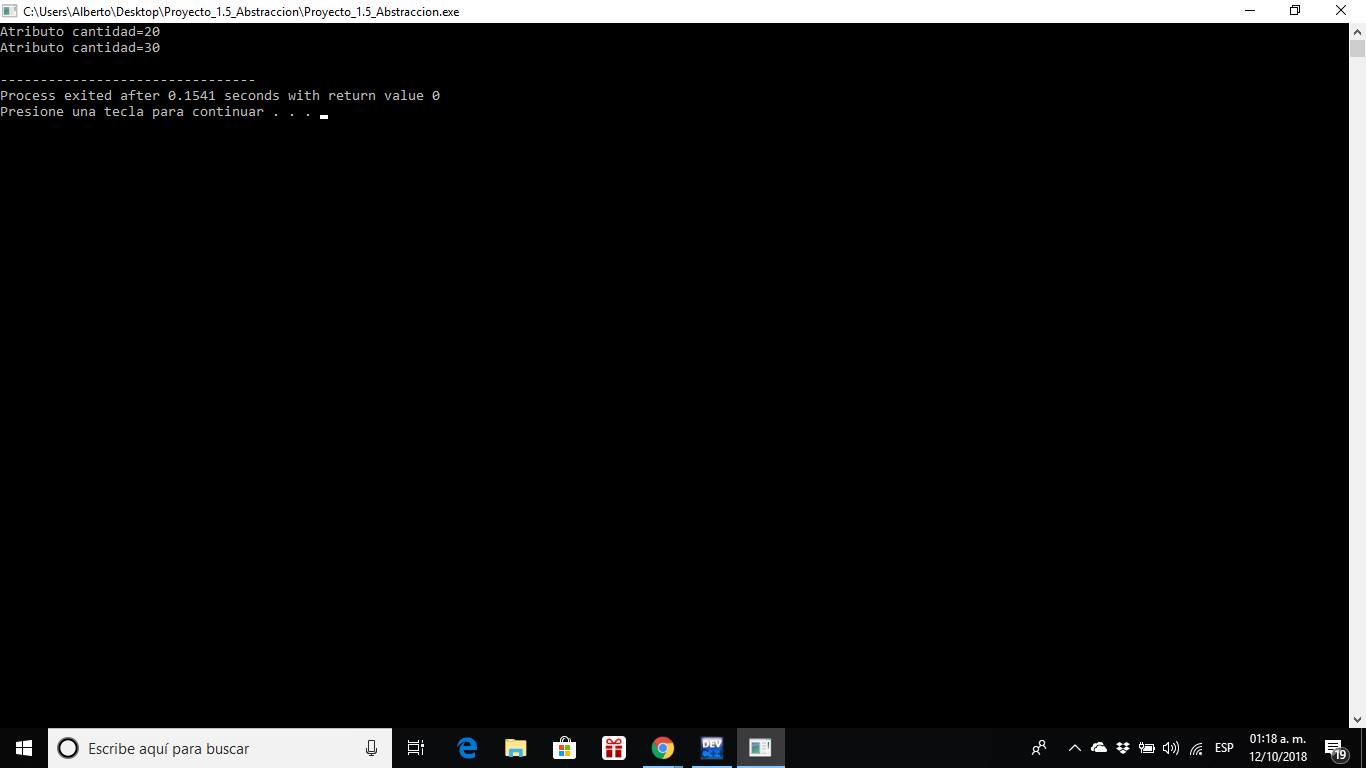
int ClaseEjAbstr::get\_cantidad(){

//return cantidad;

return cantidadArr[0];

}

**Evidencias**

****

**Conclusión**

La abstracción en programación es una herramienta muy útil al estar haciendo el programa, ya que nos ayuda a una mejor comprensión del programa, ya que ocupamos la abstracción para ocupar los conceptos ‘cotidianos’ para algunos conceptos de programación y hacer más amigable el código para las personas que lo observen