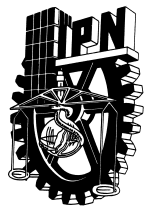
** INSTITUTO POLITECNICO NACIONAL**

**UNIDAD PROFESIONAL INTERDISCIPLINARIA**

**EN INGENIERIA Y TECNOLOGIAS**

**AVANZADAS**

Fecha: 05/12/18

Unidad de Aprendizaje: Programación Avanzada

Practica 4

Alumnos

Salgado Ríos Oscar David

García Morales Pablo

Ruíz Méndez Néstor Alberto:

Grupo: 2MV3

Profesor: Maza Casas Lamberto

**Introduccion**

Una abstracción puede definirse como: las características específicas de un objeto, aquellas que lo distinguen de los demás tipos de objetos y que logran definir límites conceptuales respecto a quien está haciendo dicha abstracción del objeto.

Una abstracción se enfoca en la visión externa de un objeto, separa el comportamiento específico de un objeto, a esta división que realiza se le conoce como la barrera de abstracción, la cual se consigue aplicando el principio de mínimo compromiso.

(Mínimo compromiso: Se refiere al proceso por el cuál la interfaz de un objeto muestra su comportamiento específico y nada más, absolutamente nada más.)

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

Pero, existe también el principio de mínima sorpresa, en el cuál una abstracción obtiene el comportamiento completo de algún objeto y por ende no ofrece sorpresas o efectos laterales que lleguen más allá del ámbito de la abstracción.

Hay una alta gama de abstracciones que existen desde los objetos que modelan muy cerca de entidades, a objetos que no tienen razón para existir:

• 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 ellas 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).

• Abstracción de coincidencia: Un objeto que almacena un conjunto de operaciones que no tienen relación entre sí.

**Desarrollo**

#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];

}

**Conclusion**

En este programa entendimos porque la abstracción es una de las herramientas más importantes y útiles que hemos visto al momento amenos para nosotros ya que hace más amigable el código a la vista y delimita el comportamiento de las clases y de los que indiquen estas a la hora de ocuparlas.