

Actividad 2 - Conexión y Tablas.

Lenguajes de Programación II.

Ingeniería en Desarrollo de Software

Tutor: Miguel Ángel Rodríguez Vega

Alumno: Ramón Ernesto Valdez Felix

Fecha: 24/11/2023

Índice

Introducción	3
Descripción	3
Justificación	4
Desarrollo	4
Conexión	5
Tablas	5
Código	5
Conclusión	7
Referencias	8

Introducción

En la actividad 2 de la materia de lenguajes de programación II, se necesita realizar un programa en la herramienta de programación de visual estudio con la compatibilidad de c++ donde requerimos la creación de un script en donde utilizaremos clases y polimorfismo a la empresa de nombre UNI donde pueda controlar algunos datos de sus empleados y alumnos con sus datos personales y se correlaciones con las subclases a utilizar para obtener la información deseada por la empresa UNI usando como ejemplo las siguientes: (Persona, alumno y maestro) estas deben de utilizar atributos que es correlaciones entre si mostrando la información completa al ser solicitada en el código script de C++, adiciona se la evidencia de la creación del programa y sus resultado para ser entregados en la actividad 2 de la materia en curso.

Descripción

En esta actividad 2 entregaremos el documento realizado de nombre “Conexión y tablas” esto nos dará el derecho a ser calificada para así obtenerte la puntuación de la calificación final de la materia impartida por el docente o maestro asignado a la materia de leguajes de la programación II, ya que es necesario realizar la documentación para la actividad 2 donde se nos pide que trabajemos con la creación de un script de c++ en la herramienta de programación de visual estudio utilizando una clase padre, subclases utilizando para obtener el requerimiento solicitado por la empresa UNI para así tener el control de sus empleados, alumnos donde esto proporcione la información correcta al realizar ejecución del script y así poder tomar las capturas de evidencia que se anexar en el documento a entregar.

Justificación

En esta actividad se trabajará con la herramienta de visual estudio con la creación de un script programado C++ donde utilice una clase maestras, tres o más subclases y contenga polimorfismos con la finalidad que se hereden los atributos principales en las subclases a utilizar en la documentación a presentar al docente de la materia de lenguajes de la programación:

- Usar una clase padres con herencia en los atributos a utilizar en las subclases.
- Subirlo al GitHub el documento realizado compartiendo el link para que pueda consultar el docente o maestro.
- Descargar e instala visual estudio para la creación del script de C++ a utilizar.
- Script de C++ como evidencia de la programación realizada para la actividad.
- Pantallas de evidencia del código y resultados de ejecución del programa de C++.

Desarrollo:

En esta actividad realizaremos las actividades siguientes un programa en el lenguaje de programación de C++ utilizando la herramienta de visual estudio donde plasmaremos el código o script solicitado por la actividad en curso de la clase de lenguaje de programación II.

Link: [GitHub](#)

Conexión.

Tabla.

Código.

En esta actividad tomaremos todas las evidencias que sean requeridas para la explicación del funcionamiento correcto del programa realizado en C++, el código del programa servirá como evidencia principal, las imágenes de ejecución evidencian de funcionalidad el código generado para la entrega de la actividad solicitada:

Código C++ utilizando como base una clase padre y 3 subclases con la herencia de os atributos a través del uso del polimorfismo en el script realizado.

```

#include <iostream>
#include <stdlib.h>
#include <string>
using namespace std;

class Persona {
private:
    string Nombre;
    int Edad;
public:
    Persona(string, int);
    virtual void Mostrar();
};

class Alumno :public Persona {
private:
    float CalFinal;
    string Nivel;
public:
    Alumno(string, int, float, string);
    void Mostrar();
};

class Maestro :public Persona {
private:
    string Materia;
public:
    Maestro(string, int, string);
    void Mostrar();
};

class Rector :public Persona {
private:
    string Uni;
public:
    Rector(string, int, string);
    void Mostrar();
};

Persona::Persona(string _Nom, int _Edad) {
    Nombre = _Nom;
    Edad = _Edad;
}

void Persona::Mostrar() {
    cout << "\tNombre: " << Nombre << endl;
    cout << "\tEdad: " << Edad << endl;
}

Alumno::Alumno(string _Nom, int _Edad, float _CalFin, string _Nvl) : Persona(_Nom, _Edad) {
    CalFinal = _CalFin;
    Nivel = _Nvl;
}

void Alumno::Mostrar() {
    Persona::Mostrar();
    cout << "\tCalificacion Final Alumno: " << CalFinal << endl;
    cout << "\tNivel Escolar: " << Nivel << endl;
}

Maestro::Maestro(string _Nom, int _Edad, string _Materia) : Persona(_Nom, _Edad) {
    Materia = _Materia;
}

void Maestro::Mostrar() {
    Persona::Mostrar();
    cout << "\tMateria Impartida: " << Materia << endl;
}

Rector::Rector(string _Nom, int _Edad, string _Uni) : Persona(_Nom, _Edad) {
    Uni = _Uni;
}

void Rector::Mostrar() {
    Persona::Mostrar();
    cout << "\tRector de la universidad: " << Uni << endl;
}

int main() {
    Persona* Vector[5];

    Vector[0] = new Alumno("Angela Duarte", 19, 80, "Primero Grado");
    Vector[1] = new Alumno("Toni Cocas", 21, 70, "Tercer Grado");
    Vector[2] = new Maestro("Jaime Zapata", 35, "Programacion C++");
    Vector[3] = new Rector("Dulce Lopez", 40, "UMI Coppel");

    Vector[0]->Mostrar();
    cout << "\n";
    Vector[1]->Mostrar();
    cout << "\n";
    Vector[2]->Mostrar();
    cout << "\n";
    Vector[3]->Mostrar();
    cout << "\n";

    system("pause");
    return 0;
}

```

En el script del programa realizado en el lenguaje de programación C++:

- Clase Padre Persona: (Nombre y Edad)
 - Subclase Alumnos.
 - Calificación Final.
 - Nivel de estudio.
 - Subclase Maestro.
 - Materia Impartida.
 - Subclase Rector.
 - Atributo: Universidad asignada.

Donde la información principal es heredad desde la clase padre a las subclases utilizadas polimorfismo en el script programado.

Ejecución de código de C++ utilizando en el programa realizado en la herramienta de visual estudio.

```

Calificación Final Alumno: 80
Nivel Escolar: Primero Grado

Nombre: Toni Cocas
Edad: 21
Calificación Final Alumno: 70
Nivel Escolar: Tercer Grado

Nombre: Jaime Zapata
Edad: 35
Materia Impartida: Programacion C++

Nombre: Dulce Lopez
Edad: 40
Rector de la universidad: UMI Coppel

"pausa" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\Users\ramon.valdez\source\repos\Poli...fismo1\Debug\Poli...fismo1.exe (proceso 24500) se cerró con el
código 0.
Para cerrar automáticamente la consola cuando se detiene la depuración, habilite Herramientas -> Opciones ->
Depuración -> Cerrar la consola automáticamente al detenerse la depuración.
Presione cualquier tecla para cerrar esta ventana. . .
  
```

Se realiza la ejecución mostrando los 4 datos solicitados con los atributos siguientes: Nombre y Edad en la Clase padre de nombre Persona, subclase Alumno, maestro y rector: con los atributos de calificación final, nivel de estudio, materia y rector de la universidad.

Conclusión

En conclusión: El sistema más idóneo para resolver un problema es descomponerlo en módulos más sencillos y luego, mediante diseños descendentes y refinamiento sucesivo, llegar a módulos fácilmente codificables. Estos módulos se deben codificar con las estructuras de control de programación estructurada. Las características más importantes de la programación orientada a objetos son: abstracción, encapsulamiento, herencia, polimorfismo, clases y objetos. Una clase es un tipo de dato definido por el usuario que sirve para representar objetos del mundo real. Un objeto de una clase tiene

dos componentes: un conjunto de atributos y un conjunto de operaciones. Un objeto es una instancia de una clase. Herencia es la relación entre clases que se produce cuando una nueva clase se crea utilizando las propiedades de una clase ya existente. Polimorfismo es la propiedad por la que un mensaje puede significar cosas diferentes dependiendo del objeto que lo recibe.

Referencias

GitHub: Let's build from here. (n.d.)

(N.d.). Retrieved November 29, 2023, from Wordpress.com website:

<https://eduarmandov.files.wordpress.com/2017/05/datastructures-ds-en-c.pdf>