**PROGRAMACIÓN APLICADA A LA AUTOMATIZACIÓN**

**PREPRACTICA # 7 Desarrollo de aplicaciones en C++ empleando herencia de clases.**

**CAPÍTULO DEL CURSO:** Programación orientada a objetos y estructuras de datos aplicadas a automatización.

**NOTA**

**TEMA DE LA ACTIVIDAD:** Creación de programas que usen clases y la aplicación de herencia de clases

**OBJETIVOS DE APRENDIZAJE:**

* Elaborar algoritmos que empleen herencia de clases para la creación de programas en C++ que resuelvan problemas prácticos.
* Usar el ambiente de desarrollo integrado de Qt Creator para la edición, compilación, ejecución y depuración de programas en C++.

**DURACIÓN:** 120 minutos

**MATERIALES Y HERRAMIENTAS:**

* Computadora con Qt Creator.

**TRABAJO PREVIO:**

* Haber realizado la prepráctica.

**INTRODUCCIÓN**

La herencia permite definir una clase modificando una o mas clases ya existentes. Estas modificaciones consisten habitualmente en añadir nuevos miembros (variables, funciones), a la clase que se está definiendo, aunque también se puede redefinir variables o funciones miembro ya existentes. La clase de la que se parte en este proceso recibe el nombre de clase base, y la nueva clase que se obtiene se denomina clase derivada.

La clase base directa de una clase derivada es la clase de la que hereda directamente la clase derivada.

**DESCRIPCIÓN DE LA PRÁCTICA:** El estudiante debe probar su correcta ejecución, si fuese necesario corregir los errores, y llegar a conclusiones sobre esta implementación.

**PROCEDIMIENTO:**

1. Crear una carpeta en el escritorio que tenga de nombre “practica\_7” donde guardará sus programas y luego podrá llevárselos con usted. (Si se hace en su PC propia puede obviar este paso, pero debe saber dónde se guardarán sus proyectos.)
2. Abra Qt Creator.
3. Implemente y pruebe la solución de la pregunta editando su programa en un fichero .cpp. Copie el código del programa al informe de esta práctica (como texto) y la(s) ventana(s) de visualización dónde se vea su ejecución de forma válida (como imagen).
4. Informe al profesor cuando haya concluido.
5. Describa las conclusiones y recomendaciones de la práctica.
6. Suba el informe a Aula Virtual.

**Ejercicio 1:**

Utilizando el ejemplo tratado en la conferencia sesión teórica anterior.

|  |  |
| --- | --- |
| **class** Figure{  **protected**:  **float** x,y;  **public**:  Figure(**float**=0, **float**=0);  **void** setPoint(**float**, **float**);  **float** getX() **const** {**return** x;};  **float** getY() **const** {**return** y;};  **float** area() **const**;  **void** print() **const**;  }; | Figure::Figure(**float** a, **float** b) {  x = a;  y = b;  }  **void** Figure::setPoint(**float** a, **float** b) {  x = a;  y = b;  }  **float** Figure::area() {  return 0;  }  **void** Figure::print() {  cout<<"Point=["<< getX() <<","<< getY()<<”]”<<endl;  } |
| **class** Circle : **public** Figure{  **protected**:  **float** radius;  **public**:  Circle(**float**=0, **float**=0, **float**=0);  **void** setRadius(**float** r)  {if (r>=0) **radius** = r;}  **float** getRadius() **const** { **return** radius;}  **float** area() **const**;  **void** print() **const**;  }; | Circle::Circle(**float** a, **float** b, **float** r)  :Figure(a, b) {  radius = r;  }  **float** Circle::area() **const** {  **return** radius\*radius\*3.14159;  }  **void** Circle::print() {  Figure::print();  cout<< "Radius = "<<getRadius()<< endl;  } | |

Defina dos clases derivadas a partir de **Circle**:

1. la primera que represente una **esfera**.
2. La segunda que represente un **cilindro**.
3. Programe las operaciones para calcular el área para ambas.
4. Cree una operación en cada clase que permita calcular el volumen.

Elabore un programa donde se creen objetos de las clases cilindro y esfera, se inicialicen sus datos a través de teclado y se imprima en pantalla: sus datos o atributos, el valor de sus áreas y el valor de sus volúmenes.

**IMPLEMENTACIONES Y RESULTADOS**

**Ejercicio #1:**

TEXTO

**CONCLUSIONES Y RECOMENDACIONES**

**CONCLUSION**

**RECOMENDACIÓN**

**BIBLIOGRAFIA**

[1]. Deitel and Deitel. Como programar en C/C++. Segunda edición o superior.

[2]. García de Jalón, J.; y otros. Aprenda C++ como si estuviera en primero. Universidad de Navarra.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| **INICIAL** | **EN DESARROLLO** | **DESARROLLADO** | **EXCELENTE** |

**Firma del Profesor**