

IE-0217 Estructuras de Datos Abstractas y Algoritmos para Ingeniería

Curso Semestral Virtual

Tarea 02 / 2 -2022

Uso de jerarquía y composición de clases con polimorfismo

Planteamiento del problema

Se va a construir una composición y jerarquía de objetos usando C++ para el manejo de información de generación de energía eléctrica, debe asignar para cada clase de C++ un archivo header y otro de código que cumpla con las siguientes especificaciones:

1. Se debe definir una class de C++ denominada *fuentes_generadoras* que contiene un campo para el *nombre_fuente* y un método polimórfico privado que debe imprimir la *ubicación*, este método debe devolver un dato que describe la región a que pertenece una fuente de generación. Los datos son privados. Por lo que debe asignar los valores en el constructor y contar con métodos para cambiarlos. El nombre de la fuente puede ser, por ejemplo: hidroeléctrica, térmica, geotérmica, solar, eólica. Debe, además contar con un método *mostrar_ubicación* que invoca al método polimórfico.
2. Se debe definir una class de C++ denominada *unidad_generadora* que hereda de la clase *fuentes_generadoras* y que adicionalmente debe contar con un campo para el *nombre_generador* (ej: gen1, unidad2, equipo3, etc.), otro para el *sitio_ubicacion* y otro campo para la *potencia_placa* (ej: 30MW, 20MW). Los atributos son privados. Por lo que debe asignar los valores en el constructor y contar con métodos para cambiarlos. La *sitio_ubicacion* solo se asigna en el constructor y no puede cambiarse. Debe ser constante. Esta variable es la que se despliega en el método polimórfico.
3. Se debe definir una class de C++ denominada *planta_generadora*. Esta debe tener un espacio para el nombre de la planta generadora, así como una lista (STL LIST) *unidades_generadoras* que contiene objetos de tipo *unidad_generadora*. También debe recibir en el constructor el lugar donde se ubica de la planta. Los campos son privados. Por lo que debe asignar los valores en el constructor y contar con métodos para cambiarlos. El dato del sitio donde se ubica es interno y solo se podrá asignar hacia el constructor de cada unidad generadora antes de insertarse en la lista. Además, debe aportar los métodos para insertar y mostrar los componentes en la lista de *unidades_generadoras*.
4. Se debe definir una class de C++ denominada *Sistema_Electrico* que contiene una lista para ingresar las plantas generadoras (STL LIST). Los atributos son privados. Por lo que debe asignar los valores en el constructor y contar con métodos para cambiarlos. En el caso de la lista,

deben aportar un método para acceder a la lista y proceder a ingresar los datos, así como mostrarlos

5. Todas las clases indicadas anteriormente deben contar con un método *despliegue_información*, que debe brindar todos los valores de los atributos propios del objeto y en caso de contener listas hacer el despliegue de cada elemento de la lista.

Construcción metodológica de solución utilizando clases.

Deben construirse cada una de las clases (H y CPP) en forma separada.

Debe hacer un archivo principal (main.cpp) que contiene dos funciones:

1. La función APP_load es donde se construye un *Sistema_Electrico*. Deben llenar (estáticamente) la lista con un grupo de 7 plantas de generación y estas pueden tener 1, 2 o 3 generadores. Las plantas deben cubrir los diversos tipos indicados en el punto 1. Todos los datos deben ser asignados.
2. La función ShowAll es donde se debe recorrer la lista de plantas y mostrar todos los detalles de cada planta y sus respectivos generadores brindando todos los datos mediante el método *despliegue_información*. Se debe mostrar la *ubicación* en que está cada generador de cada planta.

La fecha de entrega es el jueves 29 de setiembre 2022.