IE-0217 Estructuras de Datos Abstractas y Algoritmos para Ingeniería

Curso Semestral Virtual

Práctica dirigida

Cálculo de presupuesto de construcción.

Planteamiento del problema

La empresa Soluciones Inmobiliarias S.A. se dedica la construcción de desarrollos inmobiliarios y le solicita desarrollar una aplicación de cálculos preliminares de presupuestos para edificios de apartamentos.

En este ejercicio se solicita realizar el desarrollo de una jerarquía de clases que permite determinar el presupuesto de un grupo de edificios de tipo condominio habitacional según las características que se enumeran a continuación:

1. Se requiere que la jerarquía de *clases* se realice mediante composición de objetos, utilizando "LISTAS" para que almacenen los datos de contenido variable según se va detallando en los siguientes puntos.

2. El inmobiliario es un edificio compuesto por *N* pisos. (Una lista de pisos).

3. Cada piso a su vez está compuesto por total de *M* apartamentos (Una lista).

4. Cada apartamento tiene dos dimensiones: largo, ancho, no se ocupa el alto ya que el costo está dado en metro cuadrado y el alto es constante para todos los apartamentos de un mismo piso.

5. Para el cálculo de los costos debe determinarse primero el área efectiva para cada piso y el área total de todos los pisos. Al área total de cada piso se debe sumar un 35% de área que es el valor medio de espacio requerido para: pasillos, accesos, escaleras y ascensores.

6. En el caso del edificio bastará conocer el área total por piso y sumarla para tener el área efectiva por edificio. Además debe agregarse al total de apartamentos de cada edificio un espacio adicional de 24 metros cuadrados correspondiente a dos parqueos en un área separada destinada a este fin. El costo del área de parqueo es el 75% del costo del metro cuadrado de construcción.

7. El costo del metro cuadrado es de $350.00 se pretenden construir 3 edificios con las siguientes características:

    a. Edificio 1:

i. Apartamentos piso 1 (largo y ancho)
   - Apto 1 {20,15}
   - Apto 2 {20,15}
   - Apto 3 {20,15}
   - Apto 4 {20,15}
   - Apto 5 {20,15}
   - Apto 6 {20,15}
   - Apto 7 {20,15}
   - Apto 8 {20,15}
   - Apto 9 {20,15}
   - Apto 10 {20,15}
   - Apto 12 {20,15}

ii. Apartamentos piso 2 (largo y ancho)
   - Apto 1 {20,15}
   - Apto 2 {20,15}
   - Apto 3 {20,15}
   - Apto 4 {20,15}
   - Apto 5 {20,15}
   - Apto 6 {20,15}
   - Apto 7 {20,15}
   - Apto 8 {20,15}
   - Apto 9 {20,15}
   - Apto 10 {20,15}
   - Apto 12 {20,15}

iii. Apartamentos piso 3 (largo y ancho)
   - Apto 1 {20,15}
   - Apto 2 {20,15}
   - Apto 3 {20,15}
   - Apto 4 {20,15}
   - Apto 5 {20,15}
   - Apto 6 {20,15}
   - Apto 7 {20,15}
   - Apto 8 {20,15}
   - Apto 9 {20,15}
   - Apto 10 {20,15}
   - Apto 12 {20,15}
   - Apto 13 {20,15}

iv. Apartamentos piso 4 (largo y ancho)
   - Apto 1 {20,15}
   - Apto 2 {20,15}
   - Apto 3 {20,15}
   - Apto 4 {20,15}
   - Apto 5 {20,15}
   - Apto 6 {20,15}
   - Apto 7 {20,15}

- Apto 8 {20,15}
- Apto 9 {20,15}
- Apto 10 {20,15}
- Apto 12 {20,15}
- Apto 13 {20,15}

b. Edificio 2:
   i. Apartamentos piso 1 (largo y ancho)
      - Apto 1 {12,15}
      - Apto 2 {12,15}
      - Apto 3 {12,15}
      - Apto 4 {12,15}
      - Apto 5 {12,15}
      - Apto 6 {12,15}
      - Apto 7 {12,15}
      - Apto 8 {12,15}
      - Apto 9 {12,15}
      - Apto 10 {12,15}

   ii. Apartamentos piso 2 (largo y ancho)
      - Apto 1 {10,15}
      - Apto 2 {10,15}
      - Apto 3 {10,15}
      - Apto 4 {10,15}
      - Apto 5 {10,15}
      - Apto 6 {10,15}
      - Apto 7 {10,15}
      - Apto 8 {10,15}
      - Apto 9 {10,15}
      - Apto 10 {10,15}
      - Apto 11 {10,15}
      - Apto 12 {10,15}

   iii. Apartamentos piso 2 (largo y ancho)
      - Apto 1 {10,15}
      - Apto 2 {10,15}
      - Apto 3 {10,15}
      - Apto 4 {10,15}
      - Apto 5 {10,15}
      - Apto 6 {10,15}
      - Apto 7 {10,15}
      - Apto 8 {10,15}
      - Apto 9 {10,15}
      - Apto 10 {10,15}
      - Apto 11 {10,15}
      - Apto 12 {10,15}

iv. Apartamentos piso 1 (largo y ancho)
    - Apto 1 {12,15}
    - Apto 2 {12,15}
    - Apto 3 {12,15}
    - Apto 4 {12,15}
    - Apto 5 {12,15}
    - Apto 6 {12,15}
    - Apto 7 {12,15}
    - Apto 8 {12,15}
    - Apto 9 {12,15}
    - Apto 10 {12,15}

c. Edificio 3:
    i. Apartamentos piso 1 (largo y ancho)
        - Apto 1 {20,15}
        - Apto 2 {20,15}
        - Apto 3 {20,15}
        - Apto 4 {20,15}
        - Apto 5 {20,15}
        - Apto 6 {20,15}
        - Apto 7 {20,15}
        - Apto 8 {20,15}
        - Apto 9 {20,15}
        - Apto 10 {20,15}
        - Apto 12 {20,15}

    ii. Apartamentos piso 2 (largo y ancho)
        - Apto 1 {20,15}
        - Apto 2 {20,15}
        - Apto 3 {20,15}
        - Apto 4 {20,15}
        - Apto 5 {20,15}
        - Apto 6 {20,15}
        - Apto 7 {20,15}
        - Apto 8 {20,15}
        - Apto 9 {20,15}
        - Apto 10 {20,15}
        - Apto 12 {20,15}

    iii. Apartamentos piso 3 (largo y ancho)
        - Apto 1 {20,15}
        - Apto 2 {20,15}
        - Apto 3 {20,15}
        - Apto 4 {20,15}
        - Apto 5 {20,15}
        - Apto 6 {20,15}
        - Apto 7 {20,15}
        - Apto 8 {20,15}

- Apto 9 {20,15}
- Apto 10 {20,15}
- Apto 12 {20,15}
- Apto 13 {20,15}

Construcción metodológica de solución utilizando clases.

Con la finalidad de facilitar el desarrollo se dan las siguientes recomendaciones:

1. Construir un tipo (typedef) denominado T_apartamento que contiene las dimensiones de cada apartamento: largo y ancho (tipo float).

2. Construir la clase Piso que contiene:
   a. una lista privada de apartamentos (puede ser la estructura _cola vista en clase o usar STL).
   b. El total de área del piso y el área efectiva (un 35% más que la efectiva).
   c. Deben contener un método de inserción de cada apartamento en la lista.
   d. Un método que calcule el área total y el área efectiva y la almacene internamente
   e. Un método para devolver el área efectiva.
   f. Un método para devolver el área total del piso.
   g. Un método que retorne el total de apartamentos en la lista interna.

3. Construir una clase Edificio que contiene:
   a. Una lista privada de tipo Piso, en que cada componente contiene cada uno de los pisos del punto 2 (anterior).
   b. Atributos para el total de área del edificio y el área efectiva (un 35% más que la efectiva) de todo el edificio.
   c. Deben contener un método de inserción de cada piso en la lista interna
   d. Un método que calcule el área final efectiva para los valores en el punto 3.b.
   e. Un método que calcule el área total (sin 35%) para los valores en el punto 3.b.
   f. Un método que calcule el área del terreno (tomando como referencia el área del primer piso de cada edificio).

4. Desarrollar una clase BaseHabitacional con las siguientes características:
   a. Un método virtual que calcule el área total de un complejo de edificios (no presente en esta clase).
   b. Un método virtual que calcule el costo final de un complejo de edificios (no presente en esta clase).

5. Desarrollar una clase Habitacional con las siguientes características:
   a. Hereda a la clase BaseHabitacional en forma pública.

b. Tiene una lista de edificios que contiene los elementos del punto 3.
c. Un método que inserta un edificio en la lista
d. Un método protegido que calcula el total de área efectiva y el costo de la construcción completa (Edificios y parqueos).
e. Un método protegido que calcule el área total de un complejo de edificios (polimorfismo hacia BaseHabitacional).
f. Un método virtual que calcule el costo final de un complejo de edificios (polimorfismo hacia BaseHabitacional).

6. Deben construir un archivo CPP y H para cada clase enunciada. Las clases de BaseHabitacional y Habitacional pueden ir en un único par de archivos CPP y H.

7. Deben hacer un archivo main.cpp que contiene una función evaluar presupuesto que recibe en forma referencial un objeto BaseHabitacional que calcula el área y los costos finales.

8. En la función principal (main) se deben colocar en forma directa los valores indicados para cada largo y ancho de cada apartamento. No usar *cin* o *scanf* por la cantidad de datos.

9. Deben construir los objetos de clases/tipos: apartamentos, pisos, edificios y habitacional, además hacer las inclusiones para que toda la información quede en un objeto Habitacional que debe ser llamado *ComplejoHabitacional*.

10. Deben hacer el cálculo de área y costo totales desde el objeto ComplejoHabitacional, el cual deberá llamar a cada componente de su lista y a su vez los objetos contenidos deben hacer lo mismo de manera que se calcule el costo y área total a partir de este llamado.

11. La función en el main para evaluar el costo (presupuesto) y el área total debe llamar a las funciones virtuales para obtener los resultados solicitados ya que las funciones heredadas son protegidas.

12. Deben generar correctamente el código dentro los archivos C++ y H para que el contenido sea claro.