

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота №** 3

з дисципліни “Об'єктно орієнтоване програмування”

тема “Структурні шаблони”

|  |  |  |
| --- | --- | --- |
| Виконав(ла)  студент(ка) II курсу  групи КП-01  Северин Юрій Юрійович  (*прізвище, ім’я, по батькові*)  Варіант № 14 | Київ 2021 | Перевірила  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Заболотня Тетяна Миколаївна  (*прізвище, ім’я, по батькові*) |

**Мета роботи**

Ознайомитися зі структурними шаблонами у C# та реалізувати дві задачі за допомогою цих паттернів.

**Постановка завдання**

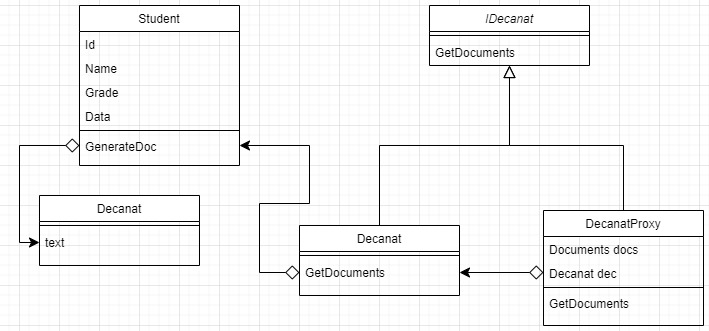
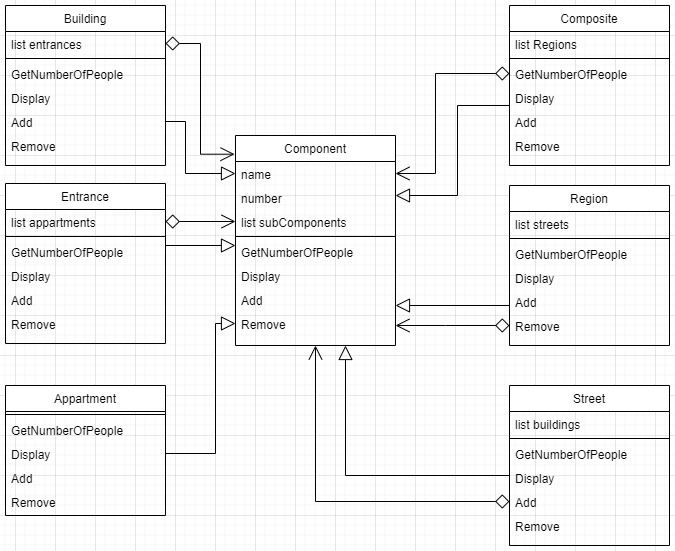
1. За допомогою шаблону проєктування реалізувати модель мікрорайону. Наприклад, район повинен складатися з вулиць, вулиці містять будинки; будинки складаються з під’їздів та квартир. Забезпечити виведення на екран інформації про кожний тип об’єктів (кількість населення вулиці, будинку тощо). Також забезпечити можливість додавання та видалення об’єктів всіх типів. Крім того, реалізувати метод «Переписати населення», який передбачає підсумовування населення мікрорайону шляхом обчислення населення вкладених в нього об’єктів.
2. Розробити модуль до програмного забезпечення, що використовується у приймальній комісії вузу, який буде автоматично створювати папку з бланками електронних документів абітурієнта (анкета, оцінки на іспитах тощо) тільки у разі, якщо останній набрав прохідний бал.

**Текст коду програми**

|  |
| --- |
| **1** |
| using System.Collections.Generic;  namespace System  {  class Client  {  static void Main()  {  Composite root = new Composite("City");  Component region = new Region("Bronx");  Component street = new Street("Main ave");  Component building = new Building("42");  Component entrance1 = new Entrance("entr3");  Component appartment1 = new Appartment("apart69", 4);  Component entrance2 = new Entrance("entr2");  Component appartment2 = new Appartment("apart694", 10);  root.Add(region);  region.Add(street);  street.Add(building);  building.Add(entrance1);  building.Add(entrance2);  entrance1.Add(appartment1);  entrance2.Add(appartment2);  root.Display();  Console.WriteLine($"\nIn city lives: {AllPeople(root)}\nIn {street.name} lives: {street.GetNumberOfPeople()}");  Console.WriteLine($"In {entrance1.name} lives: {entrance1.GetNumberOfPeople()}\nIn {entrance2.name} lives: {entrance2.GetNumberOfPeople()}");  Console.ReadKey();  }  public static int AllPeople(Component root)  {  return root.GetNumberOfPeople();  }  }  class Composite : Component  {  List<Component> Regions = new List<Component>();  public Composite(string name)  : base(name)  { }  public override void Add(Component component)  {  Regions.Add(component);  }  public override void Remove(Component component)  {  Regions.Remove(component);  }  public override void Display()  {  Console.WriteLine(name);  foreach (Component component in Regions)  {  Console.Write(" ");  component.Display();  }  }  public override int GetNumberOfPeople()  {  int sum = 0;  foreach(var a in Regions)  {  sum += a.GetNumberOfPeople();  }  return sum;  }  }  } |
| using System.Collections.Generic;  namespace System  {  internal class Appartment : Component  {  int peopleLives;  public Appartment(string name, int people = 0)  : base(name)  { peopleLives = people; }  public override void Display()  {  Console.WriteLine(name);  }  public void ChangeLivings(int c)  {  peopleLives = c;  }  public override void Remove(Component component)  {  }  public override int GetNumberOfPeople()  {  return peopleLives;  }  public override void Add(Component c)  {  }  }  } |
| using System.Collections.Generic;  namespace System  {  internal class Building : Component  {  List<Component> entrances = new List<Component>();  public Building(string name)  : base(name)  { }  public override void Display()  {  Console.WriteLine(name);  foreach (Component component in entrances)  {  Console.Write(" ");  component.Display();  }  }  public override void Add(Component component)  {  entrances.Add(component);  }  public override void Remove(Component component)  {  entrances.Remove(component);  }  public override int GetNumberOfPeople()  {  int sum = 0;  foreach (var a in entrances)  {  sum += a.GetNumberOfPeople();  }  return sum;  }  }  } |
| using System.Collections.Generic;  namespace System  {  abstract class Component  {  public string name { get; private set; }  public int number { get; private set; }  public List<Component> subComponents = new List<Component>();  public Component(string name)  {  this.name = name;  }  public abstract int GetNumberOfPeople();  public abstract void Display();  public abstract void Add(Component c);  public abstract void Remove(Component c);  }  } |
| using System.Collections.Generic;  namespace System  {  internal class Entrance : Component  {  List<Component> appartments = new List<Component>();  public Entrance(string name)  : base(name)  { }  public override void Display()  {  Console.WriteLine(name);  foreach (Component component in appartments)  {  Console.Write(" ");  component.Display();  }  }  public override void Add(Component component)  {  appartments.Add(component);  }  public override void Remove(Component component)  {  appartments.Remove(component);  }  public override int GetNumberOfPeople()  {  int sum = 0;  foreach (var a in appartments)  {  sum += a.GetNumberOfPeople();  }  return sum;  }  }  } |
| using System.Collections.Generic;  namespace System  {  class Region : Component  {  List<Component> streets = new List<Component>();  public Region(string name)  : base(name)  { }  public override void Display()  {  Console.WriteLine(name);  foreach (Component component in streets)  {  Console.Write(" ");  component.Display();  }  }  public override void Add(Component component)  {  streets.Add(component);  }  public override void Remove(Component component)  {  streets.Remove(component);  }  public override int GetNumberOfPeople()  {  int sum = 0;  foreach (var a in streets)  {  sum += a.GetNumberOfPeople();  }  return sum;  }  }  } |
| using System.Collections.Generic;  namespace System  {  internal class Street : Component  {  List<Component> buildings = new List<Component>();  public Street(string name)  : base(name)  { }  public override void Display()  {  Console.WriteLine(name);  foreach (Component component in buildings)  {  Console.Write(" ");  component.Display();  }  }  public override void Add(Component component)  {  buildings.Add(component);  }  public override void Remove(Component component)  {  buildings.Remove(component);  }  public override int GetNumberOfPeople()  {  int sum = 0;  foreach (var a in buildings)  {  sum += a.GetNumberOfPeople();  }  return sum;  }  }  } |

|  |
| --- |
| **2** |
| namespace System  {  class DecanatProxy : IDecanat  {  Documents docs;  Decanat dec;  public DecanatProxy()  {  docs = new Documents();  }  Documents IDecanat.GetDocuments(int number)  {  if (docs != null)  {  docs = new Documents();  }  if (dec == null)  {  dec = new Decanat();  }  try  {  docs = dec.GetDocuments(number);  }  catch  {  Console.WriteLine("Cannot create documents, marks are too low");  }  return docs;  }  }  } |
| namespace System  {  public class Documents  {  public Documents()  {  }  public string text;  public Documents(string document)  {  this.text = document;  }  }  } |
| namespace System  {  class Program  {  static void Main(string[] args)  {  IDecanat document = new DecanatProxy();  Student s1 = new Student("Ivan Ivanov", 69);  Student s2 = new Student("Petr Petrov", 42);  var doc1 = document.GetDocuments(1);  Console.WriteLine(doc1.text + "\n");  var doc2 = document.GetDocuments(2);  Console.WriteLine(doc2.text);  Console.Read();  }  }  interface IDecanat  {  Documents GetDocuments(int number);  }  class Decanat : IDecanat  {  public Documents GetDocuments(int id)  {  if (Student.students[id - 1].Grade > 60)  {  return Student.students[id - 1].GenerateDoc();  }  throw new Exception();  }  }  } |
| using System.Collections.Generic;  namespace System  {  class Student  {  public static List<Student> students = new List<Student>();  public Student()  {  }  public Student(string name, float grade)  {  Id = students.Count + 1;  Name = name;  Grade = grade;  students.Add(this);  }  public Student(string name, float grade, object data)  {  Id = students.Count + 1;  Name = name;  Grade = grade;  Data = data;  students.Add(this);  }  public Documents GenerateDoc()  {  return new Documents(Id + "|" + Name + "|" + Grade);  }  public int Id { get; set; }  public string Name { get; private set; }  public float Grade { get; private set; }  public object Data { get; set; }  }  } |

**UML діаграма**



**Висновки**

Ознайомитилися зі структурними шаблонами у C# та реалізували дві задачі за допомогою цих паттернів.