

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 №** 4

з дисципліни “Програмування 1. Об'єктно-орієнтоване програмування та шаблони проєктування”

тема “ C# .Net. Породжуючі шаблони у мові C#”

|  |  |  |
| --- | --- | --- |
| Виконала  студентка IІ курсу  групи КП-02  Гудзіцька Дарина Сергіївна |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Заболотня Тетяна Миколаївна  (*прізвище, ім’я, по батькові*) |

Київ 2021

**Мета роботи**

Ознайомитися з такими можливостями мови програмування C# як породжуючі шаблони проєктування, набути розуміння як і коли їх використовувати.

**Постановка завдання**

1. За допомогою шаблону проєктування реалізувати процес випуску книжок за заготовленими раніше матрицями. Кожна книга характеризується іменем автора, видавництвом, де вона була надрукована, серією та номером видання. Масив сторінок є тим об’єктом, який необхідно розтиражувати для кожної книги.
2. За допомогою шаблону проєктування реалізувати процес формування штучної водойми, який складається з таких етапів як побудова котловану, заповнення його водою, висадження рослин, запуск водних мешканців. В залежності від типу ділянки котлован може бути як бетонним (каркасним), так і виконаний як пластикова чаша чи просто яма, застелена плівкою.

**Обгрунтування вибору шаблону**

**Завдання 1. Prototype**

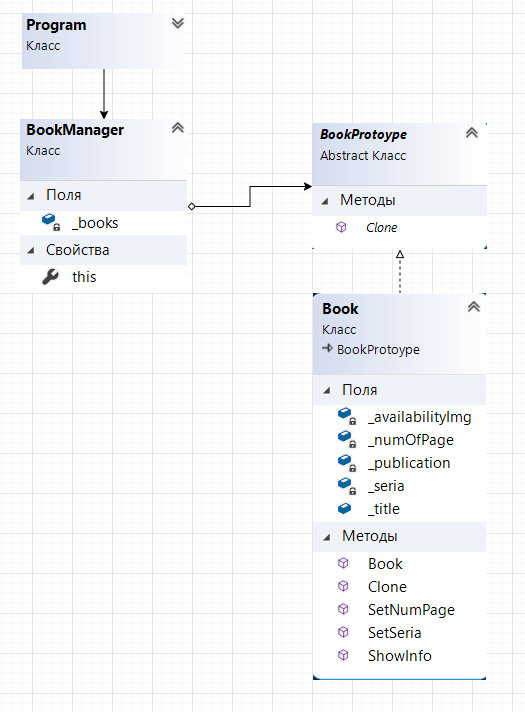
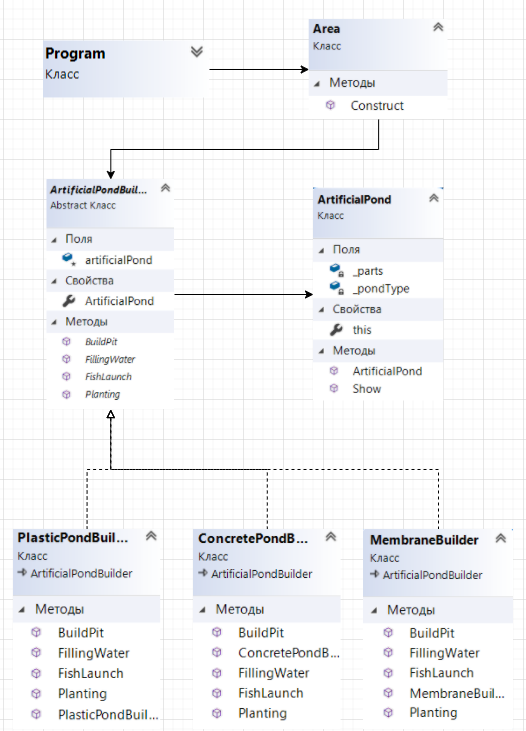
Щоб розтиражувати книжку потрібно її копіювати, при цьому деякі її характеристики (поля) змінюються, а деякі можуть бути приватними. Тому найкращим рішенням буде клонувати книжку через інтерфейс, що не прив’язується до конкретного класу книжки.

**Завдання 2. Builder**

Формування штучної водойми, це складний та багатоетапний процес. Щоб не ускладнювати код класами для всіх можливих варіантів водойми, можна винести конструювання водойми за межі його класу та доручити це окремим об’єктам, «будівникам».

**UML діаграма класів**

**Завдання 1. Завдання 2.**

****

**Фрагменти коду**

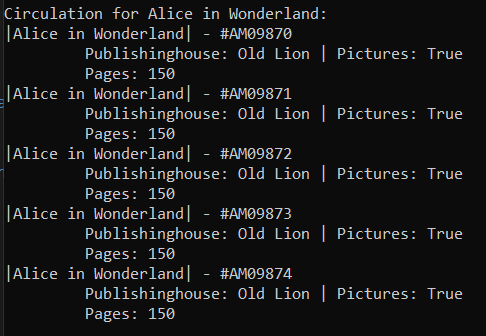
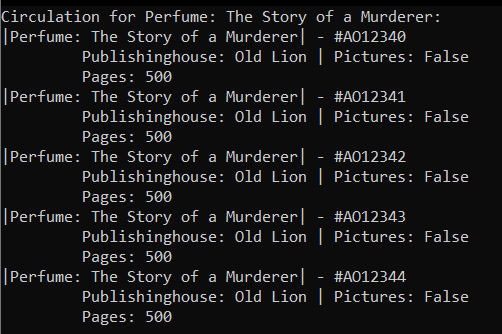
**Завдання 1.**

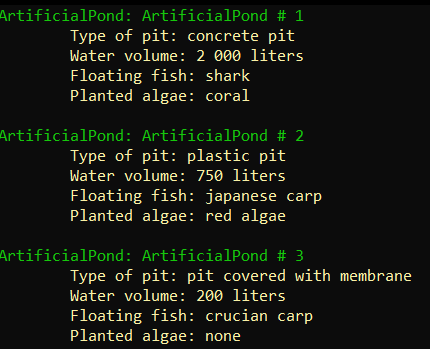
|  |
| --- |
| using System;  using System.Collections.Generic;  using static System.Console;  namespace task\_1  {  abstract class BookProtoype  {  public abstract BookProtoype Clone();  }  class Book : BookProtoype  {  public string \_title;  private string \_seria;  private bool \_availabilityImg;  private string \_publication;  private int \_numOfPage;  public Book(string t, string s, bool img, string p)  {  this.\_title = t;  this.\_seria = s;  this.\_availabilityImg = img;  this.\_publication = p;  }  public void SetNumPage(int n)  {  \_numOfPage = n;  }  public void SetSeria(string n)  {  \_seria = n;  }  public override BookProtoype Clone()  {  return this.MemberwiseClone() as BookProtoype;  }  public void ShowInfo()  {  WriteLine($"|{this.\_title}| — #{this.\_seria}\n\t" +  $" Publishinghouse: {this.\_publication} | Pictures: {this.\_availabilityImg}\n\t Pages: {this.\_numOfPage}");  }  }  class BookManager  {  private Dictionary<string, BookProtoype> \_colors =  new Dictionary<string, BookProtoype>();    public BookProtoype this[string key]  {  get { return \_colors[key]; }  set { \_colors.Add(key, value); }  }  }  } |

**Завдання 2.**

|  |
| --- |
| class Area  {  public void Construct(ArtificialPondBuilder artificialPondBuilder)  {  artificialPondBuilder.BuildPit();  artificialPondBuilder.FillingWater();  artificialPondBuilder.FishLaunch();  artificialPondBuilder.Planting();  }  }  class ArtificialPond  {  private string \_pondType;  private Dictionary<string, string> \_parts = new Dictionary<string, string>();  public ArtificialPond(string pondType)  {  this.\_pondType = pondType;  }  public string this[string key]  {  get { return \_parts[key]; }  set { \_parts[key] = value; }  }  public void Show()  {  ForegroundColor = ConsoleColor.Green;  WriteLine($"ArtificialPond: {\_pondType}");  ForegroundColor = ConsoleColor.Yellow;  WriteLine($"\tType of pit: {\_parts["pit"]}");  WriteLine($"\tWater volume: {\_parts["water volume"]}");  WriteLine($"\tFloating fish: {\_parts["fish"]}");  WriteLine($"\tPlanted algae: {\_parts["plants"]}\r\n");  }  }  abstract class ArtificialPondBuilder  {  protected ArtificialPond artificialPond;  public ArtificialPond ArtificialPond  {  get { return artificialPond; }  }  public abstract void BuildPit();  public abstract void FillingWater();  public abstract void Planting();  public abstract void FishLaunch();  }  class ConcretePondBuilder : ArtificialPondBuilder  {  public ConcretePondBuilder()  {  artificialPond = new ArtificialPond("ArtificialPond # 1");  }  public override void BuildPit()  {  artificialPond["pit"] = "concrete pit";  }  public override void FillingWater()  {  artificialPond["water volume"] = "2 000 liters";  }  public override void FishLaunch()  {  artificialPond["fish"] = "shark";  }  public override void Planting()  {  artificialPond["plants"] = "coral";  }  } |

**Приклади результатів**

**Завдання 1.**

**Завдання 2.**

**Висновки**

Виконавши дану лабораторну роботу відбулося ознайомлення з такими можливостями мови програмування C# як породжуючі шаблони проєктування, було набуте розуміння як і коли їх використовувати.