v

МIНIСТЕРСТВО ОСВIТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота №**2

з дисципліни “Основи об’єктно-орієнтованого програмування”

тема “ Реалізація складних структурних шаблонів проектування ”

|  |  |  |
| --- | --- | --- |
| Виконала  студентка II курсу  групи КП-83  Дзьоба Уляна Олегівна  (*прізвище, ім’я, по батькові*)  варіант №5 |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 2020 р.  викладач  Юсин Яків Олексійович  (*прізвище, ім’я, по батькові*) |

Київ 2020

**Мета роботи**

Ознайомлення з основними характеристиками шаблонів «Заступник», «Фасад», «Адаптер» та «Легковаговик», запам’ятовування поширених ситуацій, коли використання цих шаблонів є доцільним, набуття вмінь та навичок реалізації шаблонів під час створення програмного коду.

**Постановка завдання**

1. Розробити модуль до програмного забезпечення, що використовується у бібліотеці, який буде дозволяти читачу здійснювати пошук в електронному каталозі тільки за умови, якщо він не має незданих книжок на своєму абонементі.
2. Створити програмне забезпечення автоматичної генерації грамот для учасників конкурсу юних піаністів, а також дипломів лауреатів та дипломантів. Список учасників з вказаними школою та класом, а також з поміткою щодо зайнятого місця задається користувачем перед початком роботи програми.

**Вибір шаблону**

Для розв’язання першої задачі був обраний структурний шаблон проектування «Заступник» a саме його вид - заступник для аутентифікації (заступник-захисник) який використовується для перевірки правильності умов доступу до об’єкту, що відповідає поставленій задачі обмеження доступу до електронного каталогу в залежності від наявності незданих книг у користувача.

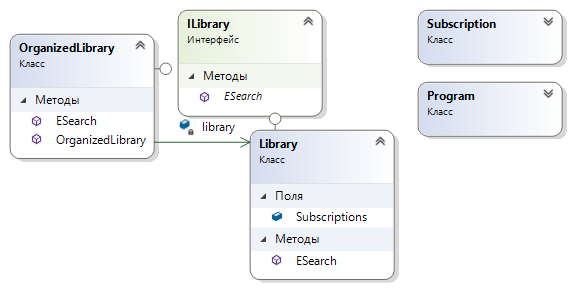
**Текст програми**

|  |
| --- |
| **Програмний код** |
| class Program  {  static void Main(string[] args)  {  List<string> books1 = new List<string> { "Odyssey", "Gone with the wind", "Great Expectations" };  List<string> books2 = new List<string> { };  Subscription sub1 = new Subscription("Ann Smith", books1);  Subscription sub2 = new Subscription("Noah Johnson", books2);  List<Subscription> subs = new List<Subscription> { sub1, sub2 };  OrganizedLibrary library = new OrganizedLibrary(subs);  library.ESearch(sub1, "Looking for Alaska");  library.ESearch(sub2, "Atomic habits");  Console.ReadKey();  }  }  class Subscription  {  public string Fullname;  public List<string> CurrentBooks;  public Subscription(string fullname, List<string>currentBooks)  {  Fullname = fullname;  CurrentBooks = currentBooks;  }  }  interface ILibrary  {  void ESearch(Subscription sub, string searchString);  }  class Library:ILibrary  {  public List<Subscription> Subscriptions;  public void ESearch(Subscription sub, string searchString)  {  Console.WriteLine("searching " + searchString);  }    }  class OrganizedLibrary:ILibrary  {  Library library = new Library();  public OrganizedLibrary(List<Subscription> subscriptions)  {  library.Subscriptions = subscriptions;  }  public void ESearch(Subscription sub, string searchString)  {  if (sub.CurrentBooks.Count == 0)  library.ESearch(sub, searchString);  else  Console.WriteLine("Return books first");  }  } |

**Приклад результатів роботи програми**

|  |
| --- |
| **Вивід у консоль** |
|  |

**Діаграма класів**

****

**Вибір шаблону**

Для розв’язання другої задачі був обраний структурний шаблон проектування «Легковаговик» оскільки згідно з теоретичними відомостями він призначений для полегшення повторного використання багатьох подібних малих об’єктів, натомість аналізуючи поставлену ситуацію було виявлено що можна спростити генерацію великої кількості подібних грамот.

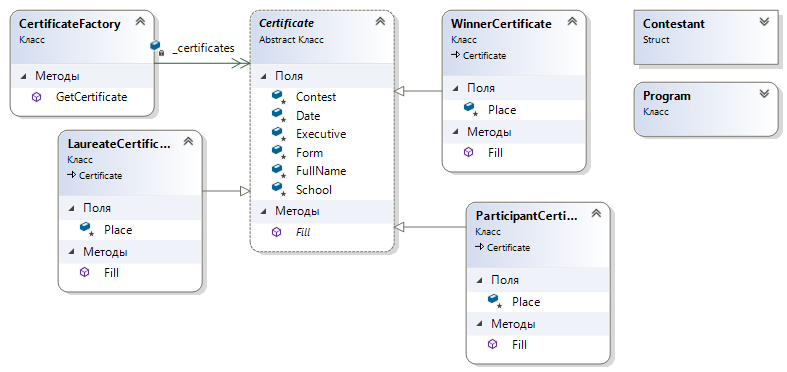
**Текст програми**

|  |
| --- |
| **Програмний код** |
| class Program  {  static void Main(string[] args)  {  List<Contestant> contestants = new List<Contestant> { new Contestant("Noah Johnson", "Echo Creek Academy", "sophomore", 3),  new Contestant("Mary Evans", "Moordale High", "senior", 2), new Contestant("Thomas Brooks", "Shermer High School", "junior", 1), new Contestant("Karen Anderson", "Newport Union", "freshman", 0),  new Contestant("Jessica Scott", "Rosewood High", "junior", 0), new Contestant("Brian Collins", "Rosewood High", "sophomore", 0) };  CertificateFactory factory = new CertificateFactory();  foreach(Contestant c in contestants)  {  Certificate certificate = factory.GetCertificate(c.Place);  certificate.Fill(c);  }  Console.ReadKey();  }  }  **class CertificateFactory**  **{**  **private Dictionary<int, Certificate> \_certificates = new Dictionary<int, Certificate>();**  **public Certificate GetCertificate(int key)**  **{**  **Certificate certificate = null;**  **if (\_certificates.ContainsKey(key))**  **{**  **certificate = \_certificates[key];**  **}**  **else**  **{**  **if(key==1) certificate = new WinnerCertificate();**  **else if(key==0) certificate = new ParticipantCertificate();**  **else certificate = new LaureateCertificate();**  **\_certificates.Add(key, certificate);**  **}**  **return certificate;**  **}**  **}**  **abstract class Certificate**  **{**  **protected string Contest="Piano Master";**  **protected string Executive="Ann Smith";**  **protected string FullName;**  **protected string School;**  **protected string Form;**  **protected string Date="14/03/20";**  **public abstract void Fill(Contestant contestant);**  **}**  **class ParticipantCertificate:Certificate**  **{**  **protected int Place;**  **public override void Fill(Contestant contestant)**  **{**  **this.FullName = contestant.FullName;**  **this.School = contestant.School;**  **this.Form = contestant.Form;**  **Console.WriteLine("\n\n\tSHCOOL'S MUSIC ASSOCIATION\n\t\t{0}\n\tCertificate of Participation\n\n This certificate is awarded to \n\t\t{1}\n {2} in {3}\n\n\t{4}\t{5}\n\t\t\tEvent Organizer\n\n",**  **this.Contest, this.FullName, this.Form, this.School, this.Date, this.Executive);**  **}**  **}**  **class LaureateCertificate:Certificate**  **{**  **protected int Place;**  **public override void Fill(Contestant contestant)**  **{**  **this.FullName = contestant.FullName;**  **this.School = contestant.School;**  **this.Form = contestant.Form;**  **this.Place = contestant.Place;**  **Console.WriteLine("\n\n\tSHCOOL'S MUSIC ASSOCIATION\n\t\t{0}\n\tAward certificate\n\n This certificate is awarded to \n\t\t{1}\n {2} in {3}\n\t\tfor {4} place\n\n\t{5}\t{6}\n\t\t\tEvent Organizer\n\n",**  **this.Contest, this.FullName, this.Form, this.School, this.Place, this.Date, this.Executive);**  **}**  **}**  **class WinnerCertificate:Certificate**  **{**  **protected int Place;**  **public override void Fill(Contestant contestant)**  **{**  **this.FullName = contestant.FullName;**  **this.School = contestant.School;**  **this.Form = contestant.Form;**  **this.Place = 1;**  **Console.WriteLine("\n\n\tSHCOOL'S MUSIC ASSOCIATION\n\t\t{0}\n\tFirst place award certificate\n\n This certificate is awarded to \n\t\t{1}\n {2} in {3}\n for the most outstanding perfomance\n\n\t{4}\t{5}\n\t\t\tEvent Organizer\n\n",**  **this.Contest, this.FullName, this.Form, this.School, this.Date, this.Executive);**  **}**  **}**  **public struct Contestant**  **{**  **public string FullName;**  **public string School;**  **public string Form;**  **public int Place;**  **public Contestant(string fullname, string school, string form, int place)**  **{**  **FullName = fullname;**  **School = school;**  **Form = form;**  **Place = place;**  **}**  **}** |

**Приклад результатів роботи програми**

|  |
| --- |
| **Вивід у консоль** |
|  |

**Діаграма класів**

****

**Висновки щодо роботи програм**

Виконавши дану лабораторну роботу було створено дві програми мовою C# що демонструють ознайомлення з використанням шаблонів проектування. Шляхом аналізу заданих ситуацій і порівняння їх характеристик з характеристиками стандартних зразків використання запропонованих шаблонів проектування для реалізації програм було обрано шаблони «Заступник» та «Легковаговик» і показано доцільність цього вибору шляхом демонстрації роботи програм.

Розробка велася за допомогою Microsoft Visual Studio.