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 №**4

з дисципліни “Основи об’єктно-орієнтованого програмування”

тема “ Реалізація поведінкових шаблонів проектування ”

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

Київ 2020

**Мета роботи**

Ознайомлення з основними характеристиками шаблонів «Стратегія», «Шаблонний метод» та «Стан», запам’ятовування поширених ситуацій, коли використання цих шаблонів є доцільним, набуття вмінь та навичок реалізації шаблонів під час створення програмного коду.

**Постановка завдання**

1. За допомогою шаблону проектування реалізувати перемикання режимів роботи сервера, до якого через мережу мають змогу підключатися різні користувачі. У випадку перевищення припустимого числа підключень, сервер перестає надавати доступ до своїх ресурсів та видає повідомлення про це всім користувачам, які роблять спроби встановити з ним з’єднання. У разі наявності можливості підключення до нього, сервер приймає запит на встановлення зв’язку.
2. За допомогою шаблона проектування реалізувати декілька комплексів спортивних занять для людей з різними фізичними можливостями: підвищеної складності, нормальний, полегшений. В залежності від фізичної підготовленості людини призначати їй той чи інший набір вправ, які вона буде виконувати під час ранкової зарядки.

**Вибір шаблону**

Для розв’язання першої задачі був обраний поведінковий шаблон проектування «Стан» оскільки згідно з теоретичними відомостями він дозволяє об'єкту варіювати свою поведінку залежно від внутрішнього стану, що відповідає поставленій задачі, в якій сервер змінює свою поведінку при перевищенні максимальної кількості користувачів.

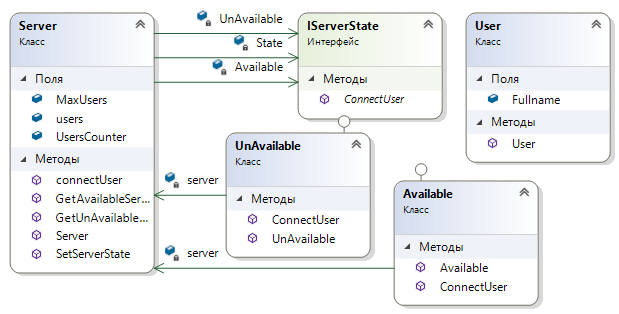
**Текст програми**

|  |
| --- |
| **Програмний код** |
| class Program  {  static void Main(string[] args)  {  Server server = new Server();  User user1 = new User("Ann Smith");  User user2 = new User("Noah Johnson");  User user3 = new User("Mary Evans");  User user4 = new User("Thomas Brooks");  User user5 = new User("James Williams");  User user6 = new User("Jessica Scott");  server.connectUser(user1);  server.connectUser(user2);  server.connectUser(user3);  server.connectUser(user4);  server.connectUser(user5);  server.connectUser(user6);  Console.ReadLine();  }  }  interface IServerState  {  void ConnectUser(User user);  }  internal class Available : IServerState  {  private Server server;  public Available(Server server)  {  this.server = server;  }  public void ConnectUser(User user)  {  server.users.Add(user);  System.Console.WriteLine("User {0} connected", user.Fullname);  server.UsersCounter++;  if (server.UsersCounter == server.MaxUsers)  {  server.SetServerState(server.GetUnAvailableServerState());  System.Console.WriteLine("Server is anavailable");  }  }  }  internal class UnAvailable : IServerState  {  private Server server;  public UnAvailable(Server server)  {  this.server = server;  }  public void ConnectUser(User user)  {  System.Console.WriteLine("Server is unavailable. Can't connect user {0}", user.Fullname);  }  }  class Server  {  IServerState Available;  IServerState UnAvailable;  IServerState State;  public int MaxUsers=5;  public int UsersCounter=0;  public List<User> users=new List<User>();  public Server()  {  this.Available = new Available(this);  this.UnAvailable = new UnAvailable(this);  State = this.Available;  if (UsersCounter > MaxUsers)  {  State = this.UnAvailable;  }  }  public void SetServerState(IServerState newServerState)  {  State = newServerState;  }  public void connectUser(User newUser)  {  State.ConnectUser(newUser);  }  public IServerState GetAvailableServerState() { return this.Available; }  public IServerState GetUnAvailableServerState() { return this.UnAvailable; }  }  class User  {  public string Fullname;  public User(string fullname)  {  Fullname = fullname;  }  }  } |

**Приклад результатів роботи програми**

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

**Діаграма класів**

****

**Вибір шаблону**

Для розв’язання другої задачі був обраний поведінковий шаблон проектування «Стратегія» оскільки згідно з теоретичними відомостями його доцільно використовувати при різних варіантах поведінки, що відповідає поставленій задачі.

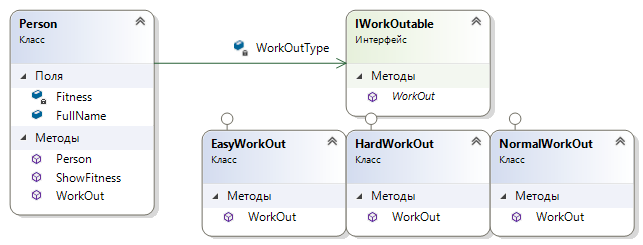
**Текст програми**

|  |
| --- |
| **Програмний код** |
| **class Program**  **{**  **static void Main(string[] args)**  **{**  **Person person1 = new Person("Ann Smith", PhysicalFitness.Fit);**  **Person person2 = new Person("Noah Johnson", PhysicalFitness.VeryFit);**  **Person person3 = new Person("James Williams", PhysicalFitness.NotFit);**  **person1.ShowFitness();**  **person1.WorkOut();**  **Console.WriteLine();**  **person2.ShowFitness();**  **person2.WorkOut();**  **Console.WriteLine();**  **person3.ShowFitness();**  **person3.WorkOut();**  **Console.ReadLine();**  **}**  **}**  **interface IWorkOutable**  **{**  **void WorkOut();**  **}**  **class EasyWorkOut : IWorkOutable**  **{**  **public void WorkOut()**  **{**  **Console.WriteLine("Doing easy work out");**  **}**  **}**  **class NormalWorkOut : IWorkOutable**  **{**  **public void WorkOut()**  **{**  **Console.WriteLine("Doing normal work out");**  **}**  **}**    **class HardWorkOut : IWorkOutable**  **{**  **public void WorkOut()**  **{**  **Console.WriteLine("Doing hard work out");**  **}**  **}**  **class Person**  **{**  **public string FullName;**  **PhysicalFitness Fitness;**  **IWorkOutable WorkOutType;**  **public Person(string name, PhysicalFitness fitness)**  **{**  **FullName = name;**  **Fitness = fitness;**  **switch (Fitness)**  **{**  **case PhysicalFitness.VeryFit:**  **WorkOutType = new HardWorkOut();**  **break;**  **case PhysicalFitness.Fit:**  **WorkOutType = new NormalWorkOut();**  **break;**  **case PhysicalFitness.NotFit:**  **WorkOutType = new EasyWorkOut();**  **break;**  **}**  **}**  **public void WorkOut()**  **{**  **WorkOutType.WorkOut();**  **}**  **public void ShowFitness()**  **{**  **Console.WriteLine(Fitness);**  **}**  **}** |

**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 «Стан» та «Стратегія» і показано доцільність цього вибору шляхом демонстрації роботи програм.

Розробка велася за допомогою Microsoft Visual Studio.