НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

"КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО"

Кафедра програмного забезпечення комп'ютерних систем

**РГР**

із дисципліни «Програмування»

Виконав:

студент 2 курсу ФПМ групи КП-93

Торговських Олександр Олегович

*(ПІБ)*

Прийняв/ла:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
|  | Бали |
| Якість виконання |  |
| Термін здачі |  |
| Сумарний бал |  |

КИЇВ — 2020

**Завдання:**

**Варіант №20**

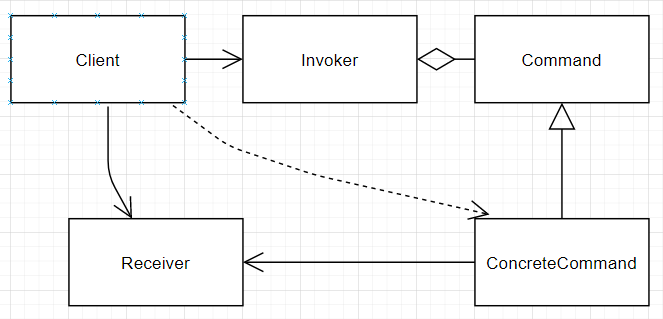
1. У вищому навчальному закладі студент є частиною навчальної групи. Група входить до складу потоку на кафедрі та курсу на факультеті в цілому. І студенти, і група, і факультет мають ім’я або назву. Також студент характеризується масивом оцінок, які він отримує протягом сесії. Потік студентів складається з масиву груп, курс – з масиву потоків, факультет – з масиву курсів. Організувати виведення оцінок студентів, згрупувавши їх по групам, потокам, курсам. Також реалізувати метод обчислення середнього балу для студента, групи тощо.
2. Розробити модуль програмної системи обліку абонентів оператора мобільного зв’язку, який при спробі абонента перейти на інший тариф буде запитувати в нього дані щодо PUK-коду телефону і відображати перелік доступних тарифів тільки у разі коректного введення цього коду.
3. За допомогою шаблона проектування забезпечити створення квитків на концерти. Друк здійснювати з урахуванням уведених даних про концерт та місця (ряд, місце у залі).
4. За допомогою шаблона проектування реалізувати декілька варіантів здійснення покупки в магазині. Процес купівлі складається з таких етапів: прихід до магазину, вибір товару, оплата, доставка. В залежності від типу магазину дані етапи можуть бути реалізовані по-різному. Етап приходу до магазину може бути як реальним, так і віртуальним (у випадку, якщо мова йдеться про Інтернет-магазин). Таким чином, для того, щоб дістатися до магазину в одному випадку треба знати його фізичну адресу, в іншому – його адресу у мережі Інтернет. Етап вибору товару в реальному магазині здійснюється за допомогою продавця-консультанта; в Інтернет-магазині – за допомогою служби підтримки сайту за телефоном або онлайн. Оплата може здійснюватися в обох видах магазинів як готівкою, так і платіжними картками. Доставка купленого товару може виконуватися як безпосередньо покупцем, так і службою доставки магазину (незалежно від його виду). Реалізувати такі варіанти процесу купівлі: покупка в інтернет-магазині з оплатою готівкою кур’єру служби доставки; покупка в Інтернет-магазині з оплатою кредитною карткою та само вивезенням товару зі складу; покупка в реальному магазині з оплатою готівкою та доставкою товару до дому покупця силами служби доставки магазину.
5. Квитки у театр не підлягають поверненню у касу. Тому у випадку, якщо у людини немає можливості піти на виставу, вона повинна намагатися перепродати квитки самостійно знайомим та друзям, або віддати рідним. У свою чергу рідні, друзі та знайомі людини можуть спробувати знайти бажаючих відвідати виставу серед своїх власних рідних, друзів та знайомих. У випадку, якщо бажаючі піти на виставу не знайдені, квитки пропадають і за них не отримують грошей. В іншому випадку продавець квитків повертає собі гроші. За допомогою шаблону проектування змоделювати ситуацію з продажем квитків до театру.

**2)**

**Обгрунтування вибору шаблону:**

Длядругого завдання я використав **шаблон “Команда”**, так як цей патерн використовується коли потрібно передавати в якості параметрів дії, які викликають відповідь що в свою чергу визначають відповідь для інших дій. Тобто в даному випадку при дії абонента (спроби переходу на новий тариф) викликається дія запиту PUK-коду, який в свою чергу перевіряється і потім в залежності від правильності введення цього коду програма вирішує подальші дії.

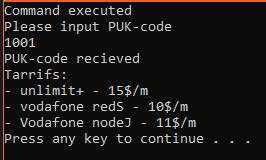
**UML-діаграма класів:**

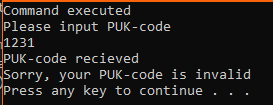
****

**Код:**

|  |
| --- |
| Program\_2.cs |
| namespace ConsoleApp1  {  class Program  {  static void Main(string[] args)  {  Client client = new Client();  client.Change\_plan();  }  }  abstract class Command  {  abstract public void Execute();  protected Receiver r;  public Receiver R  {  set  {  r = value;  }  }  }  class ConcreteCommand : Command  {  override public void Execute()  {  Console.WriteLine("Command executed");  string PUK\_code = r.Change\_plan();  if (PUK\_code != "1001")  {  Console.WriteLine("Sorry, your PUK-code is invalid");  }  else  {  Console.WriteLine("Tarrifs:\n- unlimit+ - 15$/m\n- vodafone redS - 10$/m\n- Vodafone nodeJ - 11$/m");  }  }  }  class Receiver  {  string PUK\_code;  public string Change\_plan()  {  Console.WriteLine("Please input PUK-code");  this.PUK\_code = Console.ReadLine();  Console.WriteLine("PUK-code recieved");  return PUK\_code;  }  }  class Invoker  {  private Command command;  public void StoreCommand(Command c)  {  command = c;  }  public void ExecuteCommand()  {  command.Execute();  }  }  class Client  {  public string Change\_plan()  {  Command c = new ConcreteCommand();  Receiver r = new Receiver();  c.R = r;  Console.WriteLine("Before invoker");  Invoker i = new Invoker();  i.StoreCommand(c);  i.ExecuteCommand();  return "OK";  }  }  } |

**Приклад результатів програми:**

При введенні правильного коду:

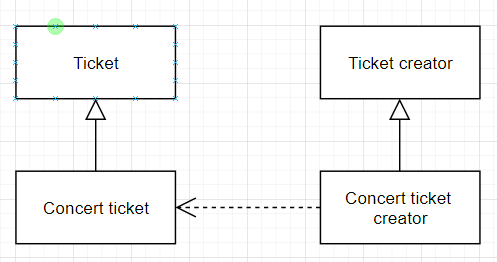
При введенні неправильного коду:

**3)**

**Обгрунтування вибору шаблону:**

Длятретього завдання я використав **шаблон “Factory Method”**, тому що в даному випадку задачею є просто створення об’єкту тип якого нам відомий зарання і система являється незалежною від створення нових об’єктів, також її можна розширяти наприклад створювати білети не тільки для концертів а і для кінотеатрів або просто театрів.

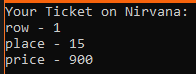
**UML-діаграма класів:**

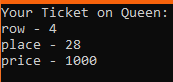
****

**Код:**

|  |
| --- |
| Program\_3.cs |
| class Program  {  static void Main(string[] args)  {  Ticket\_creator crea = new Concert\_ticket\_creator(15, "Nirvana");  Ticket ticket = crea.Create();  Console.Read();  }  }  abstract class Ticket\_creator  {  public int place;  public int row;  public string concert;  public int price;  public Ticket\_creator(int place, string concert)  {  this.place = place;  this.row = place % 7;  this.concert = concert;  if (concert == "Queen") { price = 1000; }  else { price = 900; }  }  // фабричный метод  abstract public Ticket Create();  }  class Concert\_ticket\_creator : Ticket\_creator  {  public Concert\_ticket\_creator(int place, string concert) : base(place, concert)  { }  public override Ticket Create()  {  return new Concert\_ticket(row, place, concert, price);  }  }  abstract class Ticket  { }  class Concert\_ticket : Ticket  {  public Concert\_ticket(int row, int place, string concert, int price)  {  Console.WriteLine("Your Ticket on " + concert + ":\nrow - " + row + "\nplace - " + place + "\nprice - " + price);  }  } |

**Приклад результатів програми:**





**Висновки**

Виконавши дану роботу я навчився працювати з шаблонами проектування що в більшості випадків полегшують роботу та покращують якість програми методами інкапсуляції, абстракції, наслідування та інших.