

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 “ Об'єктно-орієнтоване програмування та шаблони проєктування ”

тема “Породжуючі шаблони проєктування”

|  |  |  |
| --- | --- | --- |
| Виконав  Студент II курсу  групи КП-02  Щербатюк Ярослав Сергійович  (*прізвище, ім’я, по батькові*)  варіант №21 |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Заболотня Тетяна Миколаївна  (*прізвище, ім’я, по батькові*) |

Київ 2021

**Мета роботи**

Навчитися на практиці визначати, який породжуючий шаблон проєктування слід використовувати в залежності від умов задачі.

**Постановка завдання**

1. За допомогою шаблону проєктування реалізувати програму, яка в залежності від введеного цінового діапазону формує віртуальний комп’ютер зі складових, що підходять до заданої ціни.
2. Підприємство в Китаї випускає побутові пилососи трьох видів: для ручного прибирання, робот-пилосос та портативний пилосос для авто. Існує два види кожного з цих типів пилососів: тільки для сухого прибирання та для сухого/вологого прибирання. За допомогою шаблону проєктування реалізувати переключення виробничих ліній на випуск того чи іншого виду пилососів.

**Тексти коду програми**

|  |
| --- |
| **Builder.cs (1 задача)** |
| using System.Text;  public class Computer  {      public string Processor {get; set;}      public string VideoCard {get; set;}      public string Motherboard {get; set;}      public override string ToString()      {          StringBuilder sb = new StringBuilder();          sb.Append("Computer:\n");          sb.Append($"Processor: {this.Processor}\n");          sb.Append($"Video card: {this.VideoCard}\n");          sb.Append($"Motherboard: {this.Motherboard}\n");          return sb.ToString();      }  }  public abstract class ComputerBuilder  {      public abstract void SetProcessor();      public abstract void SetVideoCard();      public abstract void SetMotherboard();      public abstract Computer GetComputer();  }  public class CheapComputerBuilder : ComputerBuilder  {      public Computer Computer { get; private set; }      public CheapComputerBuilder()      {          this.Computer = new Computer();      }      public override void SetProcessor()      {          this.Computer.Processor = "Intel Core i3";      }      public override void SetVideoCard()      {          this.Computer.VideoCard = "NVIDIA GeForce GTX 880M 8 ГБ GDDR5";      }      public override void SetMotherboard()      {          this.Computer.Motherboard = "Asus Prime B460M-K";      }      public override Computer GetComputer()      {          return this.Computer;      }  }  public class MidleComputerBuilder : ComputerBuilder  {      public Computer Computer { get; private set; }      public MidleComputerBuilder()      {          this.Computer = new Computer();      }      public override void SetProcessor()      {          this.Computer.Processor = "Intel Core i5";      }      public override void SetVideoCard()      {          this.Computer.VideoCard = "Palit GeForce GTX 1660";      }      public override void SetMotherboard()      {          this.Computer.Motherboard = "Asus TUF Gaming Z590-Plus";      }      public override Computer GetComputer()      {          return this.Computer;      }  }  public class ExpensiveComputerBuilder : ComputerBuilder  {      public Computer Computer { get; private set; }      public ExpensiveComputerBuilder()      {          this.Computer = new Computer();      }      public override void SetProcessor()      {          this.Computer.Processor = "Intel Core i9";      }      public override void SetVideoCard()      {          this.Computer.VideoCard = "NVIDIA Quadro RTX 5000 MXM";      }      public override void SetMotherboard()      {          this.Computer.Motherboard = "Asus ROG STRIX Z690-F Gaming";      }      public override Computer GetComputer()      {          return this.Computer;      }  }  public class Assistant  {      public void Build(ComputerBuilder builder)      {          builder.SetMotherboard();          builder.SetVideoCard();          builder.SetProcessor();      }      public ComputerBuilder SelectComputer(double budget)      {          if(budget < 1000)          {              throw new System.Exception("Unable to biuld computer");          }          ComputerBuilder builder;          if(budget > 1000 && budget < 2000)          {              builder = new CheapComputerBuilder();          }          else if (budget > 2000 && budget < 4000)          {              builder = new MidleComputerBuilder();          }          else          {              builder = new ExpensiveComputerBuilder();          }          return builder;      }  } |

|  |
| --- |
| **AbstractFactory.cs (2 задача)** |
| public abstract class HandVacumCleaner  {      public string Name {get; protected set;}      public abstract void Clean();  }  public class OneModeHandVacumCleaner : HandVacumCleaner  {      public OneModeHandVacumCleaner()      {          this.Name = "One Mode Hand Vacum Cleaner";      }      public override void Clean() {}  }  public class TwoModeHandVacumCleaner : HandVacumCleaner  {      public TwoModeHandVacumCleaner()      {          this.Name = "Two Mode Hand Vacum Cleaner";      }      public override void Clean() {}      public void SwichMode() {}  }  public abstract class RobotVacumCleaner  {      public string Name {get; protected set;}      public abstract void Clean();  }  public class OneModeRobotVacumCleaner : RobotVacumCleaner  {      public OneModeRobotVacumCleaner()      {          this.Name = "One Mode Robot Vacum Cleaner";      }      public override void Clean() {}  }  public class TwoModeRobotVacumCleaner : RobotVacumCleaner  {      public TwoModeRobotVacumCleaner()      {          this.Name = "Two Mode Robot Vacum Cleaner";      }      public override void Clean() {}      public void SwichMode() {}  }  public abstract class PortableVacumCleaner  {      public string Name {get; protected set;}      public abstract void Clean();  }  public class OneModePortableVacumCleaner : PortableVacumCleaner  {      public OneModePortableVacumCleaner()      {          this.Name = "One Mode Portable Vacum Cleaner";      }      public override void Clean() {}  }  public class TwoModePortableVacumCleaner : PortableVacumCleaner  {      public TwoModePortableVacumCleaner()      {          this.Name = "Two Mode Portable Vacum Cleaner";      }      public override void Clean() {}      public void SwichMode() {}  }  public abstract class VacumCleanerFactory  {      public abstract RobotVacumCleaner CreateRobotVacumCleaner();      public abstract HandVacumCleaner CreateHandVacumCleaner();      public abstract PortableVacumCleaner CreatePortableVacumCleaner();  }  public class OneModeVacumCleanerFactory : VacumCleanerFactory  {      public override HandVacumCleaner CreateHandVacumCleaner()      {          return new OneModeHandVacumCleaner();      }      public override PortableVacumCleaner CreatePortableVacumCleaner()      {          return new OneModePortableVacumCleaner();      }      public override RobotVacumCleaner CreateRobotVacumCleaner()      {          return new OneModeRobotVacumCleaner();      }  }  public class TwoModeVacumCleanerFactory : VacumCleanerFactory  {      public override HandVacumCleaner CreateHandVacumCleaner()      {          return new TwoModeHandVacumCleaner();      }      public override PortableVacumCleaner CreatePortableVacumCleaner()      {          return new TwoModePortableVacumCleaner();      }      public override RobotVacumCleaner CreateRobotVacumCleaner()      {          return new TwoModeRobotVacumCleaner();      }  } |

**Приклади роботи програм**

|  |
| --- |
| **Задача 1** |
| PS C:\Users\Yarik Scherbatyuk\Documents\OOP\_2course\_scherbatyukk\lab4\src\1> dotnet run  Enter budget  1000  Computer:  Processor: Intel Core i9  Video card: NVIDIA Quadro RTX 5000 MXM  Motherboard: Asus ROG STRIX Z690-F Gaming  Enter budget  700  Unable to biuld computer  Enter budget  5000  Computer:  Processor: Intel Core i9  Video card: NVIDIA Quadro RTX 5000 MXM  Motherboard: Asus ROG STRIX Z690-F Gaming  Enter budget  10000  Computer:  Processor: Intel Core i9  Video card: NVIDIA Quadro RTX 5000 MXM  Motherboard: Asus ROG STRIX Z690-F Gaming  Enter budget  exit |

|  |
| --- |
| **Задача 2** |
| PS C:\Users\Yarik Scherbatyuk\Documents\OOP\_2course\_scherbatyukk\lab4\src\2> dotnet run  One Mode Robot Vacum Cleaner  One Mode Hand Vacum Cleaner  One Mode Portable Vacum Cleaner  Enter command  continue  One Mode Robot Vacum Cleaner  One Mode Hand Vacum Cleaner  One Mode Portable Vacum Cleaner  Enter command  swich  Two Mode Robot Vacum Cleaner  Two Mode Hand Vacum Cleaner  Two Mode Portable Vacum Cleaner  Enter command  swich  One Mode Robot Vacum Cleaner  One Mode Hand Vacum Cleaner  One Mode Portable Vacum Cleaner  Enter command  exit |

**UML-діаграми**

|  |
| --- |
|  |
| **Задача 1** |

|  |
| --- |
|  |
| **Задача 2** |

**Обґрунтування використання шаблонів**

1. В даній задачі використано шаблон Будівельник, тому що потрібно створювати різні варіації комп’ютера в залежності від бюджету комп’ютера. Також створення віртуального комп’ютера можна поділити на етапи, які необхідно виконати, щоб зібрати комп’ютер.
2. В даній задачі використано шаблон Абстрактна Фабрика, тому що для переключення виробничих ліній між видами цих пилососів потрібно переключатись між інтерфейсами для створення сімейств, в даному випадку переключання ліній в залежності від режимів роботи пилососів.

**Висновки**

В ході лабораторної роботи я покращив знання про породжуючі шаблони проектування. Навчився використовувати шаблони проектування в залежності від умов задачі.