МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра автоматизированных систем управления



Отчет по лабораторной работе № 6  
по дисциплине «Технологии и методы программирования»  
«Шаблоны проектирования»

Выполнили:

студент группы АВТ-214

Крипаков Леонид Дмитриевич  
Саушкин Никита Максимович

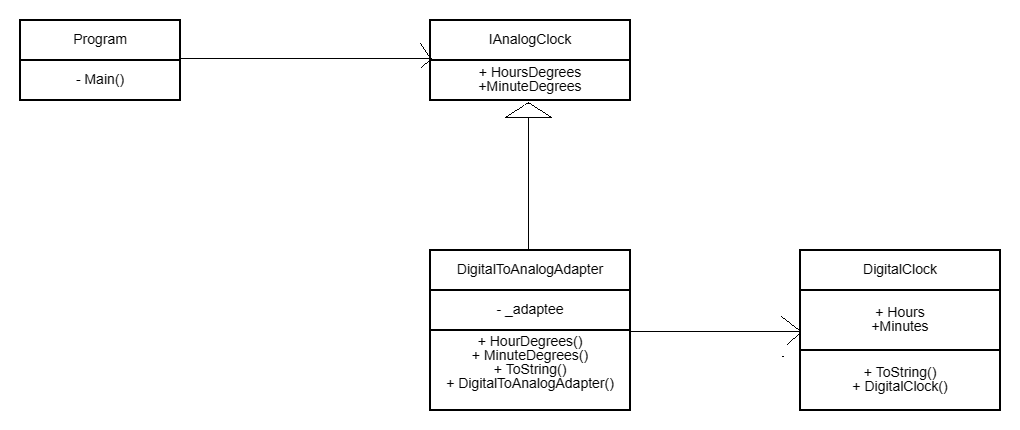
г. Новосибирск

2024 г

**ЗАДАНИЕ**

1. Познакомиться со списком известных шаблонов проектирования.  
 2. Шаблон “Адаптер”. Проект “Часы”. В проекте должен быть реализован адаптер, который дает возможность пользоваться часами со стрелками так же, как и цифровыми  
часами. В классе “Часы со стрелками” хранятся повороты стрелок.  
3. Нарисовать в UML диаграмму классов реализуемой программы (проектирование), используя любой доступный пакет, поддерживающий нотацию UML. Это может быть MS  
Visio, ArgoUML ([https://argouml.ru.uptodown.com/windows](https://vk.com/away.php?to=https%3A%2F%2Fargouml.ru.uptodown.com%2Fwindows&cc_key=)), StarUML ([https://staruml.io](https://vk.com/away.php?to=https%3A%2F%2Fstaruml.io&cc_key=)).  
4. Реализовать программу на выбранном языке программирования (реализация).

# UML-ДИАГРАММА



# ПРОГРАММА

|  |
| --- |
|  |
| using System.Text.RegularExpressions;  namespace ADigitalToAnalogAdapter  {  internal interface IAnalogClock  {  public int HourDegrees { get; set; }  public int MinuteDegrees { get; set; }  }  class DigitalToAnalogAdapter : IAnalogClock  {  private readonly DigitalClock \_adaptee;  public int HourDegrees  {  get  {  return \_adaptee.Hours % 12 \* 30;  }  set  {  \_adaptee.Hours = value \* 12 / 30;  }  }  public int MinuteDegrees  {  get  {  return \_adaptee.Minutes \* 6;  }  set  {  \_adaptee.Minutes = value / 6;  }  }  public override string ToString()  {  return $"{HourDegrees}:{MinuteDegrees}";  }  public DigitalToAnalogAdapter(DigitalClock adaptee)  {  \_adaptee = adaptee;  }  }  class DigitalClock  {  public int Hours { get; set; }  public int Minutes { get; set; }  public override string ToString()  {  return $"{Hours}:{Minutes}";  }  public DigitalClock(int hours, int minutes)  {  Hours = hours;  Minutes = minutes;  }  }  internal static class Program  {  private static void Main()  {  while (true)  {  Console.Write("Введите время в формате HH:MM в 24-часовом формате: ");  var digitalTime = Console.ReadLine();    var timeParts = digitalTime?.Split(':');  if (timeParts?.Length != 2)  {  Console.WriteLine("Введено неправильное время");  continue;  }  var checkHours = int.TryParse(timeParts[0], out var hours);  var checkMinutes = int.TryParse(timeParts[1], out var minutes);    var pattern = "^(0[0-9]|1[0-9]|2[0-3]):[0-5][0-9]$";  if (checkHours && checkMinutes && Regex.IsMatch(digitalTime!, pattern))  {  DigitalClock digitalClock = new DigitalClock(hours, minutes);  IAnalogClock analogClock = new DigitalToAnalogAdapter(digitalClock);  Console.WriteLine("Время в градусах: " + analogClock);    // analogClock.HourDegrees = 0;  // analogClock.MinuteDegrees = 180;  //  // Console.WriteLine("Время в градусах: " + digitalClock);  }  else  {  Console.WriteLine("Введено неправильное время");  continue;  }  break;  }  }  }  } |

**РЕЗУЛЬТАТЫ РАБОТЫ**

****





|  |
| --- |
| Adapter |
| class Adapter : IShowTime { AnalogClock analogClock; public Adapter(AnalogClock ac) { analogClock = ac; }  public void ShowTime() { uint hour; if (analogClock.Hour > 12) { hour = analogClock.Hour - 12; } else { hour = analogClock.Hour; }  hour \*= 30; uint minute = analogClock.Minute \* 6; uint second = analogClock.Second \* 6; hour += analogClock.Minute / 2; Console.WriteLine($"Время в градусах: {hour}:{minute}:{second}"); } |