МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ АВТОМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Кафедра Автоматизированных систем управления

**ОТЧЕТ О ВЫПОЛНЕНИИ**

**Лабораторной работы № 7**

Дисциплина «*Технологии и методы программирования*»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Выполнили:** |  | **Проверил:** |  | |
| Студенты | *Бузмаков А.И.,*  *Шестаков К.Д.* | Преподаватель | *Астапчук В. А., Пустовских Д. А.* | |
| Факультет | *АВТФ* |  |  | |
| Направление (специальность) подготовки | *09.03.01 – Информатика и вычислительная техника* | Балл: \_\_\_\_\_\_\_\_\_ | |  |
| Группа | *АП-227* | Оценка *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* | | |
| Шифр |  |  |  | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| подпись | | подпись | | |
| Дата сдачи: | «10» мая 2024г. | Дата защиты: | «\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_г. | |

Новосибирск 2024

**Текст задания**

* 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. Для установки времени цифровые часы требуют формат представления HH:MM, а “Часы со стрелками” должны получать углы поворота стрелок от их текущего положения.
  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), StarUML (https://staruml.io).
  4. Реализовать программу на выбранном языке программирования (реализация). Образцы (шаблоны) программных модулей приведены в приложениях 2 - 5.

**Результаты разработанной программы**

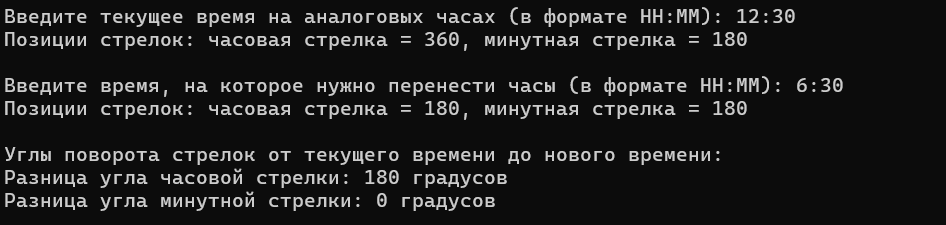


Рис. 1 - Работа программы

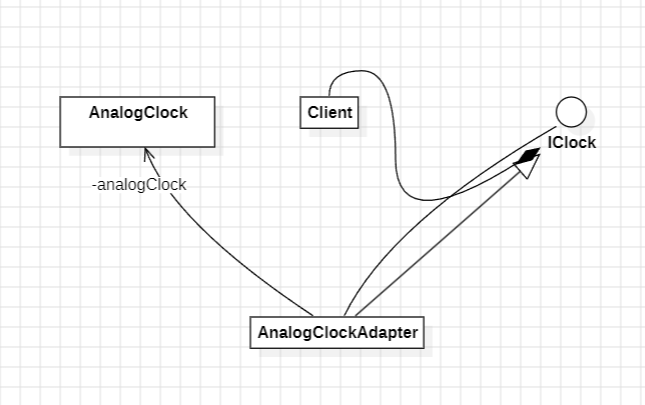


Рис. 2 - Диаграмма классов

**Текст разработанной программы**

|  |
| --- |
| Program.cs |
| namespace laba7  {  class Program  {  static void Main(string[] args)  {  // Создаем экземпляр клиента  Client client = new Client();  // Создаем часы со стрелками  AnalogClock analogClock = new AnalogClock();  // Создаем адаптер и передаем ему часы со стрелками  IClock clock = new AnalogClockAdapter(analogClock);  // Вводим текущее время на аналоговых часах  Console.Write("Введите текущее время на аналоговых часах (в формате HH:MM): ");  string currentTimeInput = Console.ReadLine();  string[] currentTimeParts = currentTimeInput.Split(':');  int currentHours = int.Parse(currentTimeParts[0]);  int currentMinutes = int.Parse(currentTimeParts[1]);  // Устанавливаем текущее время с использованием адаптера  client.SetClockTime(clock, currentHours, currentMinutes);  Console.WriteLine("");  // Запрашиваем время, на которое нужно перенести часы  Console.Write("Введите время, на которое нужно перенести часы (в формате HH:MM): ");  string newTimeInput = Console.ReadLine();  string[] newTimeParts = newTimeInput.Split(':');  int newHours = int.Parse(newTimeParts[0]);  int newMinutes = int.Parse(newTimeParts[1]);  // Устанавливаем новое время с использованием адаптера  client.SetClockTime(clock, newHours, newMinutes);  Console.WriteLine("");  // Вычисляем углы поворота стрелок от текущего времени до нового времени и выводим их  int hourAngleDifference = Math.Abs((newHours - currentHours) \* 30); // Каждый час = 30 градусов  int minuteAngleDifference = Math.Abs((newMinutes - currentMinutes) \* 6); // Каждая минута = 6 градусов  Console.WriteLine($"Углы поворота стрелок от текущего времени до нового времени:");  Console.WriteLine($"Разница угла часовой стрелки: {hourAngleDifference} градусов");  Console.WriteLine($"Разница угла минутной стрелки: {minuteAngleDifference} градусов");  }  }  } |

|  |
| --- |
| AnalogClock.cs |
| namespace laba7  {  internal class AnalogClock  {  public void SetHandsPosition(int hourHandAngle, int minuteHandAngle)  {  Console.WriteLine($"Позиции стрелок: часовая стрелка = {hourHandAngle}, минутная стрелка = {minuteHandAngle}");  }  }  } |

|  |
| --- |
| AnalogClockAdapter.cs |
| namespace laba7  {  internal class AnalogClockAdapter : IClock  {  private AnalogClock analogClock;  public AnalogClockAdapter(AnalogClock analogClock)  {  this.analogClock = analogClock;  }  public void SetTime(int hours, int minutes)  {  // Преобразуем часы и минуты в углы поворота стрелок  int hourHandAngle = hours \* 30; // В предположении, что 12 часов = 360 градусов  int minuteHandAngle = minutes \* 6; // В предположении, что 60 минут = 360 градусов  // Вызываем метод установки позиции стрелок у часов со стрелками  analogClock.SetHandsPosition(hourHandAngle, minuteHandAngle);  }  }  } |

|  |
| --- |
| Client.cs |
| namespace laba7  {  internal class Client  {  public void SetClockTime(IClock clock, int hours, int minutes)  {  clock.SetTime(hours, minutes);  }  }  } |

|  |
| --- |
| IClock.cs |
| namespace laba7  {  internal interface IClock  {  void SetTime(int hours, int minutes);  }  } |