**Министерство науки и высшего образования российской федерации**

**ФГАОУ ВО «Пермский государственный национальный исследовательский университет»**

Физический факультет

Технологии .NET Framework

Лабораторная работа №5 на тему

**«****Визуальное проектирование классов. Наследование»**

Выполнил:

студент 2 курса физического факультета

группа ПМИ-2-2021

направление «01.03.02 Прикладная математика и информатика»

профиль «Инженерия программного обеспечения»

Макарова Полина Фёдоровна

Пермь, 2023 г.

1. Цель работы

• формирование навыков визуального проектирования классов

• формирование навыков реализации наследования

1. Диаграмма классов

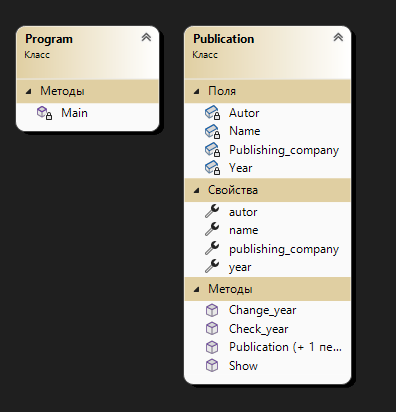


Рисунок 1 - Диаграмма классов инд.задания 1

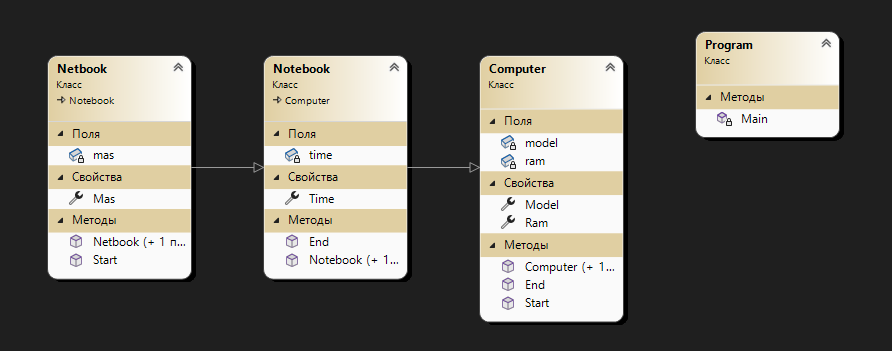


Рисунок 2 - Диаграмма классов 2 инд.задания.1.

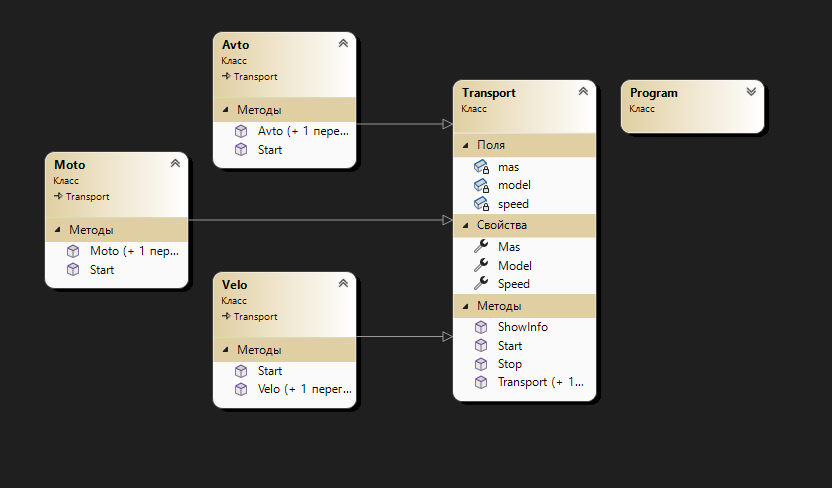


Рисунок 3 - Диаграмма классов 2 инд.задания. 2.

1. Листинг программы

Листинг программы 1 задания

|  |
| --- |
| Листинг класса Publication |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Reflection;  using System.Runtime.Intrinsics.Arm;  using System.Text;  namespace Lab\_5\_Publication  {  public class Publication  {  // Скрытые поля  // Издание - Name, Автор - Autor,  // Издательство - Publishing\_company, Год издания - Year  private string Name;  private string Autor;  private string Publishing\_company;  private int Year;  // Конструктор без параметров  public Publication()  {  this.name = "";  this.autor = "";  this.publishing\_company = "";  this.year = 0;  }  // Конструктор с параметрами  public Publication(string Name, string Autor, string Publishing\_company, int Year)  {  this.name = Name;  this.autor = Autor;  this.publishing\_company = Publishing\_company;  this.year = Year;  }  // Общедоступные свойства  public string name  {  get { return Name; }  set { Name = (value != "") ? value : "NoName"; }  }  public string autor  {  get { return Autor; }  set { Autor = (value != "") ? value : "NoName"; }  }  public string publishing\_company  {  get { return Publishing\_company; }  set { Publishing\_company = (value != "") ? value : "NoName"; }  }  public int year  {  get { return Year; }  set { Year = (value == 0) ? 1564 : value; }  }  // Общедоступные методы  // Вывод всех полей класса  public void Show()  {  Console.WriteLine("\n Издание {0}, Автор: {1}, Издательство: {2}, Год издания: {3}", name, autor, publishing\_company, year);  }  // Метод, что бы изменить год издания, увеличив или уменьшив его на указанное число лет  public void Change\_year(int N)  {  year += N;  }  // Метод, что бы определить, попадает ли год издания в заданный интервал  public void Check\_year(int a, int b)  {  if ((year < b) && (year > a))  {  Console.WriteLine("\n Год попадает в указанный интервал");  }  else  Console.WriteLine("\n Год не попадает в указанный интервал");  }  }  } |

|  |
| --- |
| Листинг класса Program |
| /\* Макарова Полина ПМИ-2  \* Вариант 10  \* Лабораторная работа 5  \*/  /\* 1. Используя инструменты и методы визуального проектирования,  \* создайте приложения, в которых определяются классы, поля, конструкторы,  \* свойства. Информация по полям выводится методом Show.  \* (public void Show()  \* { Console.WriteLine("…. {0}, {1}, {2}", param1, param2, param3); })  \* В методе Main класса Program создаются и инициализируются 2–3 объекта класса.  \* Демонстрируется ограничение недопустимых значений.  \*  \* 1. Создайте класс "Издание".  \* 2. Добавьте поля, например: название, фамилия автора, год издания.  \* 3. Для каждого поля создайте свойство с методами доступа get и set.  \* 4. Обеспечьте проверку правильности вводимых данных.  \* 5. Добавьте метод, выводящий на экран значения всех полей класса и продемонстрируйте его работу.  \* 6. Создайте три объекта данного класса. Задайте значения свойствам класса и выведите значения на экран.  \* 7. Добавьте конструктор по умолчанию и конструктор принимающий один или несколько параметров.  \* 8. Создайте объекты данного класса с помощью различных конструкторов.  \* 9. Добавьте в класс следующие методы: 1) изменить год издания, увеличив или уменьшив его на указанное число лет;  \* 2) определить, попадает ли год издания в заданный интервал.  \* 10. Обеспечьте возможность продемонстрировать правильность работы созданных методов.  \*/  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using Lab\_5\_Publication;  internal class Program  {  private static void Main(string[] args)  {  Console.SetWindowSize(75, 30);  Console.BackgroundColor = ConsoleColor.White;  Console.ForegroundColor = ConsoleColor.Black;  Console.Clear();  // Создаётся новый объект класса через конструктор по умолчанию  Console.Write("\n Создание первого объекта класса:");  Publication Publ1 = new Publication();  Publ1.Show();  // Вызов ствойств объекта  Publ1.name = "Первое";  Publ1.autor = "Соловьёв";  Publ1.publishing\_company = "Лань";  Publ1.year = 1998;  Console.Write("\n После изменения свойств объекта1:");  Publ1.Show();  // Создание других объектов класса через конструктор  Console.Write("\n Другие объекты класса 2 и 3:");  Publication Publ2 = new Publication("Пятое", "Лебедев", "Лань", 2003);  Publ2.Show();  Publication Publ3 = new Publication("Шестое", "Лебедев", "Москва", 2016);  Publ3.Show();  // Метод, чтобы изменить год издания, увеличив или уменьшив его на указанное число лет  string N;  Console.Write("\n Введите число на которое надо изменить дату: ");  N = Console.ReadLine();  Publ1.Change\_year((int)Convert.ToDouble(N));  Console.Write("\n После увеличения года издания объекта1:");  Publ1.Show();  // Метод, что бы определить, попадает ли год издания в заданный интервал  Console.Write("\nВведите поочерёдно границы интервала:");  int a, b;  Console.Write("\nМеньшая граница a = ");  N = Console.ReadLine();  a = (int)Convert.ToDouble(N);  Console.Write("\nБольшая граница b = ");  N = Console.ReadLine();  b = (int)Convert.ToDouble(N);  Console.Write("\nПроверяем попадает ли год в интервал у мервого объекта >");  Publ1.Check\_year(a, b);  }  } |

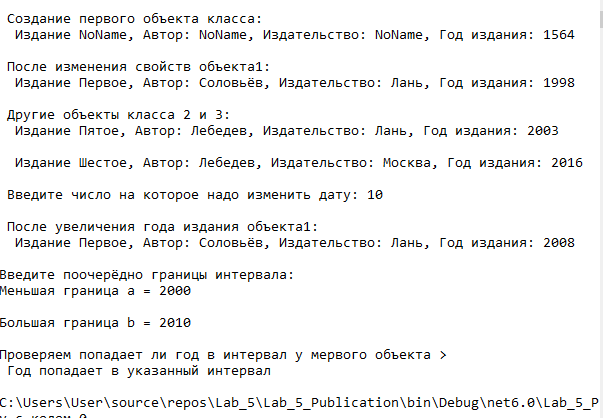


Рисунок 4 - Результат выполнения программы 1

Листинг программы 2 задания. 1.

|  |
| --- |
| Листинг класса Netbook |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace Lab\_5  {  public class Netbook : Notebook  {  private double mas;  public Netbook()  {}  public Netbook(string model, int ram, int time, double mas) : base(model, ram, time)  {  this.Mas = mas;  }  public double Mas  {  get { return mas; }  set { mas = (value < 3) ? 4.5 : value; }  }  public override void Start()  {  Console.WriteLine("{0} работает, память = {1}, масса = {2}", Model, Ram, Mas);  }  }  } |

|  |
| --- |
| Листинг класса Notebook |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace Lab\_5  {  public class Notebook : Computer  {  private int time;  // Конструктор по умолчанию  public Notebook()  {}  // Конструктор с параметрами  public Notebook(string model, int ram, int time) : base(model, ram)  {  this.Time = time;  }  // Общедоступные свойства  public int Time  {  get { return time; }  set { time = (value < 10) ? 15 : value; }  }  // Создаём новый метод End класса - наследника с модификатором override  public override void End()  {  Console.WriteLine("{0} выключается, заряд {1} мин", Model, Time);  }  }  } |

|  |
| --- |
| Листинг класса Computer |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace Lab\_5  {  public class Computer  {  // Скрытые поля  private string model;  private int ram;  // Конструктор без параметров  public Computer()  {  }  // Конструктор с параметрами  public Computer(String model, int ram)  {  this.Model = model;  this.Ram = ram;  }  // Общедоступные свойства  public string Model  {  get { return model; }  set { model = (value != "") ? value : "NoName"; }  }  public int Ram  {  get { return ram; }  set { ram = (value < 500) ? 640 : value; }  }  // Общедоступный метод End  public virtual void End()  {  Console.WriteLine("{0} выключается", Model);  }  // Общедоступный метод Start  public virtual void Start()  {  Console.WriteLine("{0} работает, память = {1}", Model, Ram);  }  }  } |

|  |
| --- |
| Листинг класса Program |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using Lab\_5;  class Program  {  static void Main(string[] args)  {  Computer comp = new Computer("IBM", 2048);  comp.Start();  comp.End();  Notebook nb = new Notebook("Asus", 1024, 120);  nb.Start(); nb.End();  Computer Comp2 = new Notebook("Del", 4096, 30);  Comp2.Start(); Comp2.End();  Netbook net = new Netbook("aboba", 2024, 90, 40);  net.Start(); net.End();  Notebook netb = new Netbook("AbObA", 4096, 70, 10);  netb.Start(); netb.End();  Console.ReadKey();  }  } |

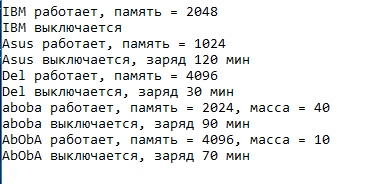


Рисунок 5 - Результат выполнения программы 2

Листинг программы 2 задания. 2.

|  |
| --- |
| Листинг класса Velo |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace Lab\_5\_two  {  public class Velo : Transport  {  public Velo()  {}  public Velo(string mode, int speed, int mas) : base(mode, speed, mas)  {}  public override void Start()  {  Console.WriteLine(" ...тишина...", Model);  }  }  } |

|  |
| --- |
| Листинг класса Moto |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace Lab\_5\_two  {  public class Moto : Transport  {  public Moto()  {}  public Moto(string mode, int speed, int mas) : base(mode, speed, mas)  {}  public override void Start()  {  Console.WriteLine(" ...ТАК-ТАК-ТАК...", Model);  }  }  } |

|  |
| --- |
| Листинг класса Avto |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace Lab\_5\_two  {  public class Avto : Transport  {  public Avto()  {}  public Avto(string mode, int speed, int mas) :base(mode, speed, mas)  {    }  public override void Start()  {  Console.WriteLine(" ...ЖЖЖЖЖ...", Model);  }  }  } |

|  |
| --- |
| Листинг класса Transport |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace Lab\_5\_two  {  public class Transport  {  private int speed;  private string model;  private int mas;  public Transport()  {}  public Transport(string model, int speed, int mas)  {  this.Model = model;  this.speed = speed;  this.mas = mas;  }  public int Mas  {  get { return mas; }  set { mas = (value < 0) ? value : 0; }  }  public string Model  {  get { return model; }  set { model = (value != "") ? value : "NoName"; }  }  public int Speed  {  get { return speed; }  set { speed= (value < 0) ? value : 0; }  }  public virtual void Start()  {  Console.WriteLine(" {0} стартует", Model);  }  public void Stop()  {  Console.WriteLine(" {0} останавливается\n", Model);  }  public void ShowInfo()  {  Console.WriteLine(" Модель: {0}, масса: {1}, скорость: {2}", Model, Mas, Speed);  }  }  } |

|  |
| --- |
| Листинг класса Program |
| /\* Макарова Полина ПМИ-2  \* Вариант 10  \* Лабораторная работа 5  \*/  /\* Используя инструменты и методы визуального проектирования, создайте классы  \* с наследниками, содержащие указанные поля, конструкторы и методы. В классе  \* Program создаются и инициализируются 2–3 объекта и указанными методами выводится  \* информация. Продемонстрируйте результаты, изменяя параметры.  \*  \* Спроектируйте класс Transport с полями mоdel (модель), speed (скорость), mas (масса)  \* и методами Start (вывод звука), Stop, ShowInfo. Наследуйте от него классы Avto (автомобиль),  \* Moto (мотоцикл), Velo (велосипед). Переопределите метод Start для каждого транспорта  \* (например, машина - …жжжжж.., мотоцикл - ..так-так-так.., велосипед - ..тишина..).  \*/  using Lab\_5\_two;  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  internal class Program  {  private static void Main(string[] args)  {  Console.SetWindowSize(75, 30);  Console.BackgroundColor = ConsoleColor.White;  Console.ForegroundColor = ConsoleColor.Black;  Console.Clear();  // Создание объекта автомобиль  Avto Av = new Avto("Lada", 60, 500);  Av.ShowInfo();  Av.Start(); Av.Stop();  // Создание объекта мотоцикл  Moto Mo = new Moto("Восход", 75, 80);  Mo.ShowInfo();  Mo.Start(); Mo.Stop();  // Создание объекта велосипед  Velo Ve = new Velo("Кама", 5, 15);  Ve.ShowInfo();  Ve.Start(); Ve.Stop();  }  } |

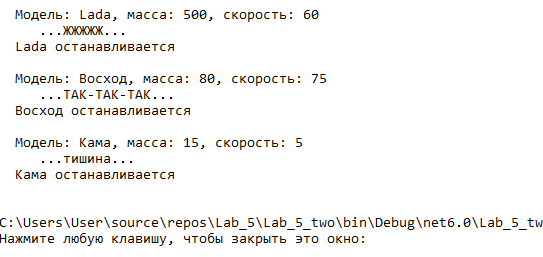


Рисунок 6 - Результат выполнения программы 3

1. Ответы на вопросы

***1. Для чего скрываются поля класса?***

Это помогает защитить данные от воздействия извне, что ведёт к меньшим возможным ошибкам в работе программы.

***2. Как можно получить доступ к скрытым полям?***

Доступ к скрытым членам можно получить только внутри тела класса или структуры, в которой они объявлены, через методы и свойства.

.

***3. Для чего и где употребляется модификатор virtual?***

Модификатор virtual используется в базовом классе и указывает, что метод может быть переопределен в производном классе.

***4. Для чего и где употребляется модификатор override?***

Модификатор override используется в производном классе и указывает новую реализацию метода, унаследованного от базового класса.