Лабораторная работа №10

**Тема:** Разработка программ, реализующих механизм наследования классов.

**Цель:** Научится разработке программ, реализующих механизм наследования классов**.**

**Техническое оснащение:** MS Windows 7, MS Word 2016, MS Visual Studio 2013, MS Visio 2019.

Выполнение работы

Вариант 24

**Пример 1.** Разработаем класс для работы с одномерным массивом. Создадим в нём индексатор, позволяющий обращаться к элементу массива по индексу.

Проверка работы примера (рисунок 1).

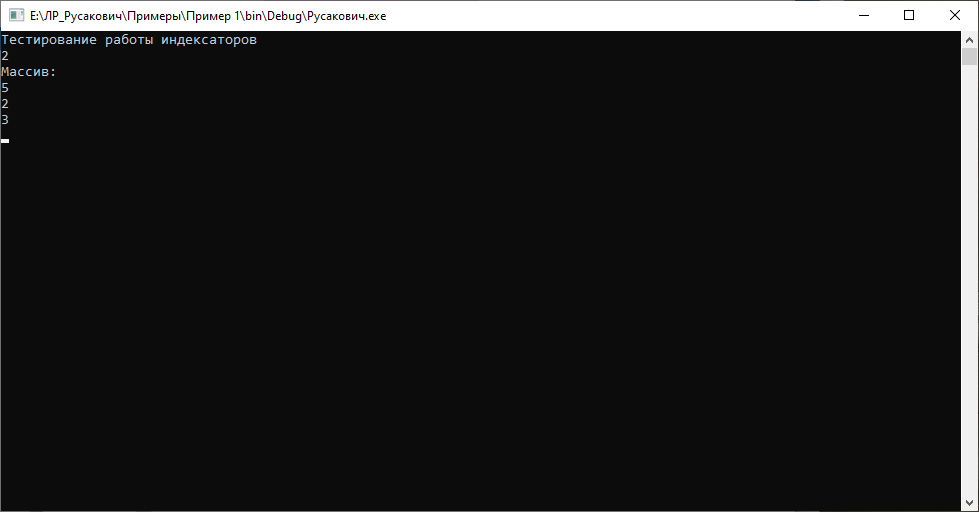


Рисунок 1 – Выполнение работы примера 1

**Пример 2.** Разработаем класс для работы с двумерным массивом (матрицей). Создадим в нём индексатор, позволяющий обращаться к элементу матрицы по индексу.

Проверка работы примера (рисунок 2).

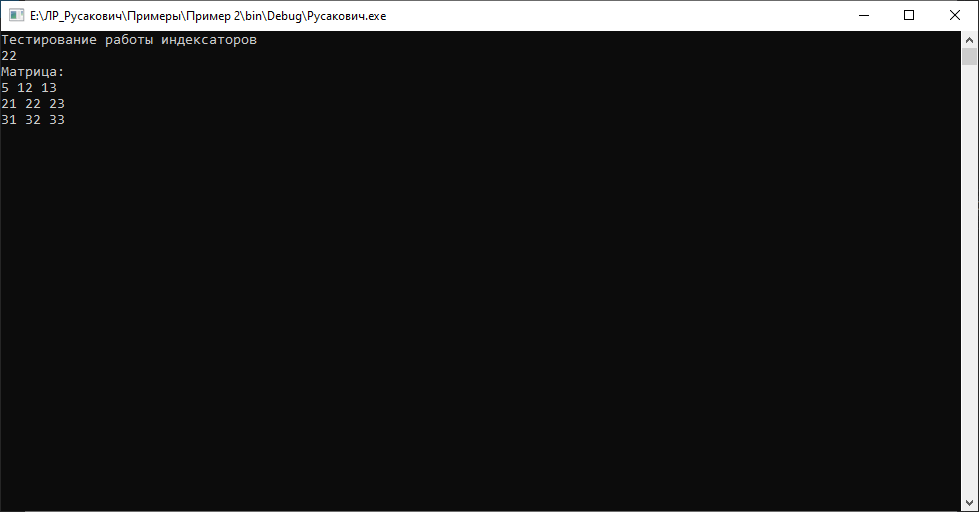


Рисунок 2 – Выполнение работы примера 2

**Пример 3.** Использование абстрактного класса.

Проверка работы примера (рисунок 3).

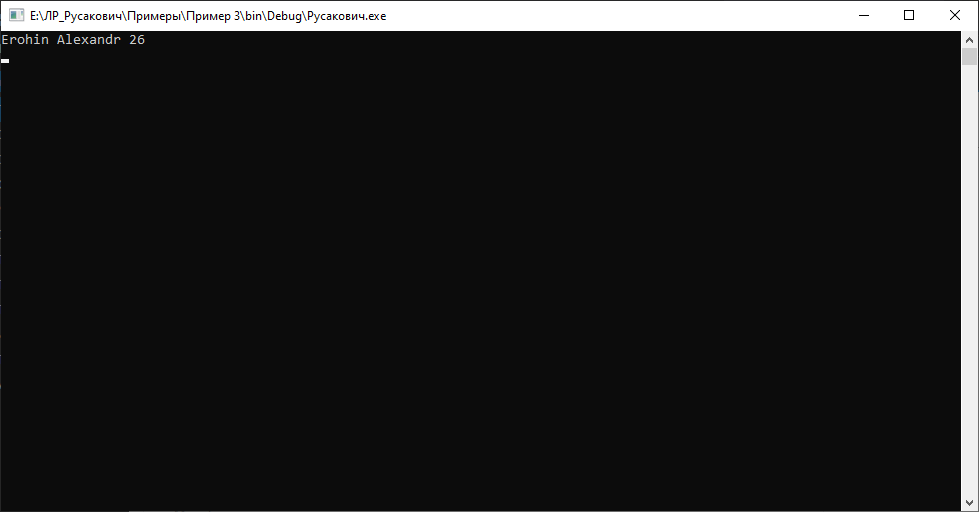


Рисунок 3 – Выполнение работы примера 3

**Пример 4.** Использование наследования

Проверка работы примера (рисунок 4).

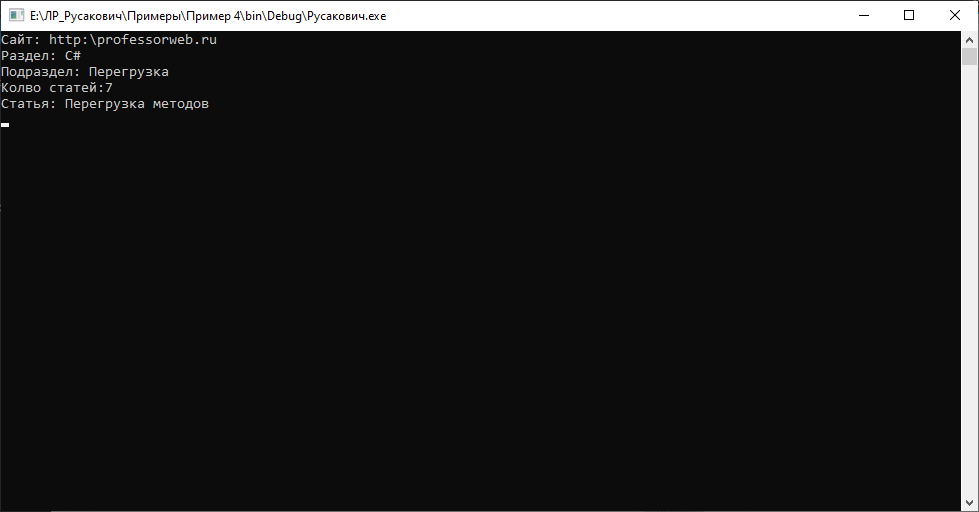


Рисунок 4 – Выполнение работы примера 4

**Пример 5.** Использование конструктора при наследовании

Проверка работы примера (рисунок 5).

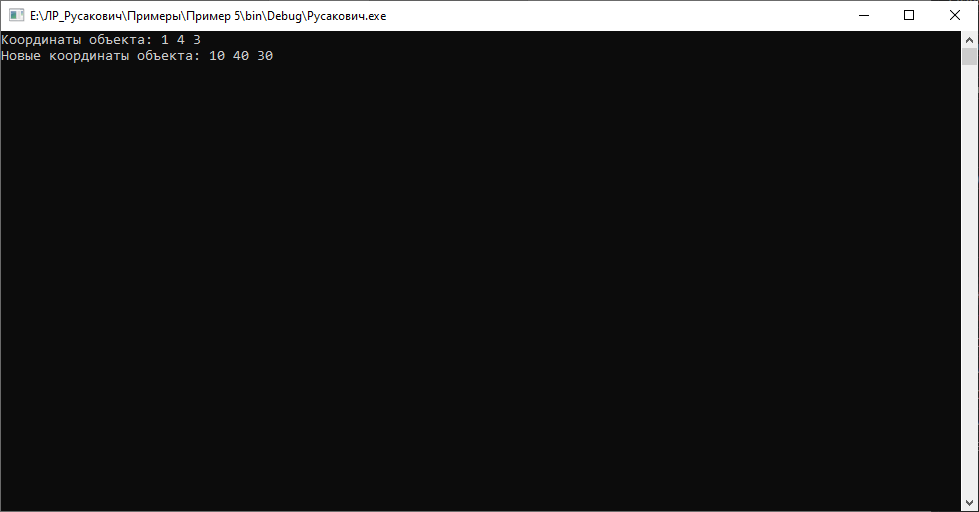


Рисунок 5 – Выполнение работы примера 5

**Пример 6.** Использование базового класса.

Проверка работы примера (рисунок 6).

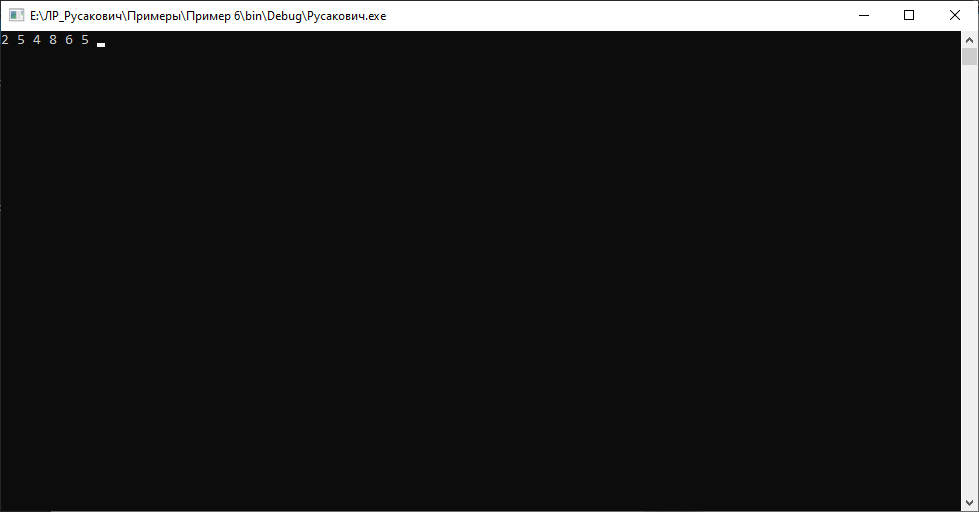


Рисунок 6 – Выполнение работы примера 6

**Пример 7.** Использование виртуальных методов, свойств и индексаторов.

Проверка работы примера (рисунок 7).

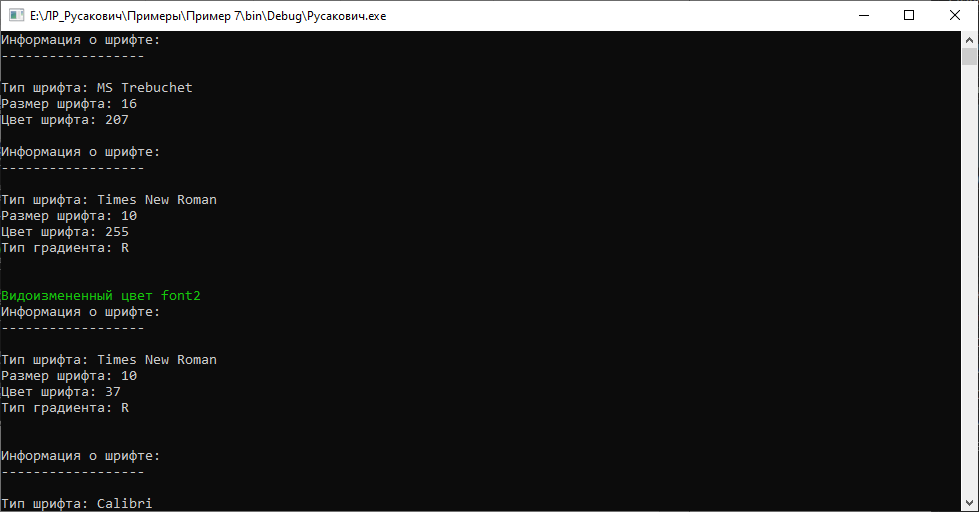


Рисунок 7 – Выполнение работы примера 7

**Задание.** Написать класс «студенческая группа». Предусмотреть возможность работы с переменным числом студентов, поиска студента по какому-либо признаку (например, по фамилии, имени, дате рождения), добавления и удаления записей, сортировки по разным полям, доступа к записи по номеру. Написать программу, демонстрирующую все разработанные элементы класса. Реализовать механизм наследования.

Блок схема алгоритма программы (рисунок 8).



Рисунок 8 ­ Блок-схема

Листинг программы:

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Threading;  namespace Bloknote  {    public class zapis  {  public int num;  public string name;  public string famil;  public DateTime dataroj;  public string numtel;  public zapis(int num, string name, string famil, DateTime dataroj, string numtel)  {  this.num = num;  this.name = name;  this.famil = famil;  this.dataroj = dataroj;  this.numtel = numtel;  }  }  public class bloknote : zapis, IComparable  {  public static List<bloknote> notzap = new List<bloknote>();  public bloknote(int num, string name, string famil, DateTime dataroj, string numtel)  : base(num, name, famil, dataroj, numtel)  {  }  public int CompareTo(object obj)  {  bloknote b = obj as bloknote;  return string.Compare(this.famil, b.famil);  }  }  public class Program  {  public int n;  public static void addzap()  {  Console.Write("---------Введите номер записи: ");  int num = Convert.ToInt32(Console.ReadLine());  Console.Write("\n-------Введите имя: ");  string name = Convert.ToString(Console.ReadLine());  Console.Write("\n-------Введите фамилия: ");  string famil = Convert.ToString(Console.ReadLine());  Console.Write("\n-------Введите дату рождения: ");  DateTime dataroj = Convert.ToDateTime(Console.ReadLine());  Console.Write("\n-------Введите номер телефона: ");  string numtel = Convert.ToString(Console.ReadLine());  bloknote zap = new bloknote(num, name, famil, dataroj, numtel);  bloknote.notzap.Add(zap);  }    private static void poisk()  {  int nomer;  Console.WriteLine("------Введите номер записи: ");  nomer = int.Parse(Console.ReadLine());  List<bloknote> ret = new List<bloknote>();  foreach (bloknote zap in bloknote.notzap) {  if (zap.num == nomer)  {  Console.Write("\t" + zap.name + " ");  Console.Write(zap.famil + " ");  Console.Write(zap.dataroj);  Console.Write(" " + zap.numtel);    }    }    }    public static void FamilSort()  {  bloknote.notzap.Sort();  foreach (bloknote i in bloknote.notzap )  {  Console.Write("\t" + i.num + " " + i.name + " " + i.famil + " " + i.dataroj + " " + i.numtel + "\n");  }  }  public static void vivod(List<bloknote> zap)  {  int count = 0;  foreach (bloknote i in bloknote.notzap)  {  Console.Write("\t"+i.num + " " + i.name + " " + i.famil + " " + i.dataroj + " " + i.numtel + "\n");  count++;  }  }  private static void DelZap()  {  int nomer;  Console.WriteLine("------Введите номер записи: ");  nomer = int.Parse(Console.ReadLine());  List<bloknote> ret = new List<bloknote>();  for (int i = 0; i < bloknote.notzap.Count; i++)  {  if(bloknote.notzap[i].num == nomer)  {  bloknote.notzap.RemoveAt(i);  }  }    }  static void Main(string[] args)  {  List<bloknote> zap1 = new List<bloknote>();  metka:  Console.WriteLine(" \t");  Console.WriteLine(" \t.........1. Добавить запись........");  Console.WriteLine(" \t.........2. Удалить запись.........");  Console.WriteLine(" \t.........3. Сортировать список.....");  Console.WriteLine(" \t.........4. Вывести список.........");  Console.WriteLine(" \t.........5. Найти запись...........");  Console.WriteLine(" \t.........6. Выход..................");  Console.WriteLine(" \t");  Console.Write("\n\t......Выберите действие ");  try  {  int n = int.Parse(Console.ReadLine());  switch (n)  {  case 1: addzap(); Console.Clear(); break;  case 2: DelZap(); Console.Clear(); break;  case 3: FamilSort(); Console.ReadKey(); Console.Clear(); break;  case 4: vivod(zap1); Console.ReadKey(); Console.Clear(); break;  case 5: poisk(); Console.ReadKey(); Console.Clear(); break;  case 6: Environment.Exit(0); break;  }  goto metka;  }  catch  {  }  Console.ReadKey(true);  }    }  } |

Результат выполнения программы приведен на рисунке 9 – 14:



Рисунок 9 – Добавление записи показано

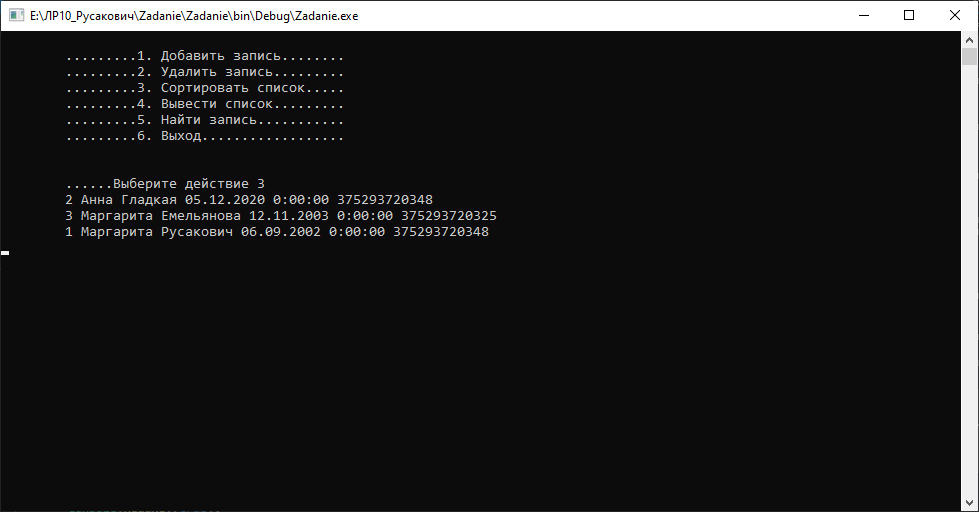


Рисунок 9 – Сортировка списка

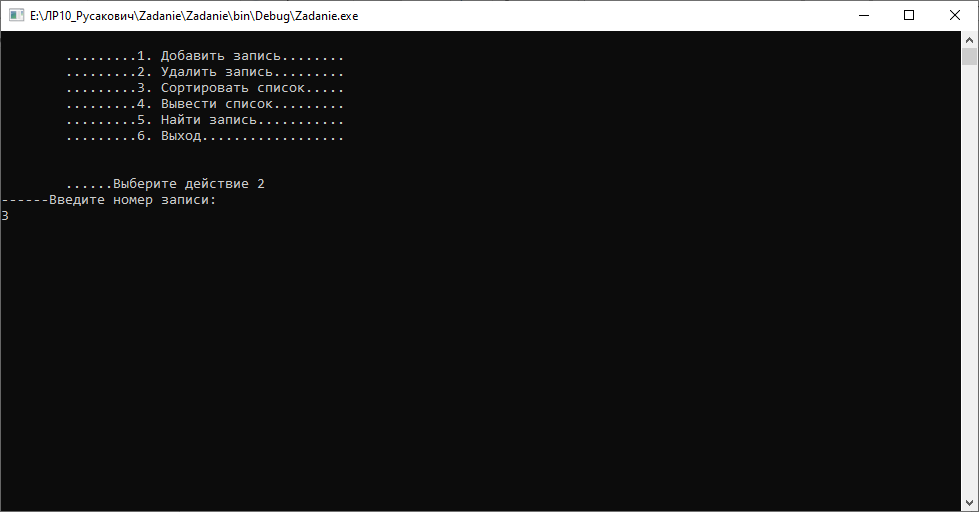


Рисунок 9 – Удаление записи

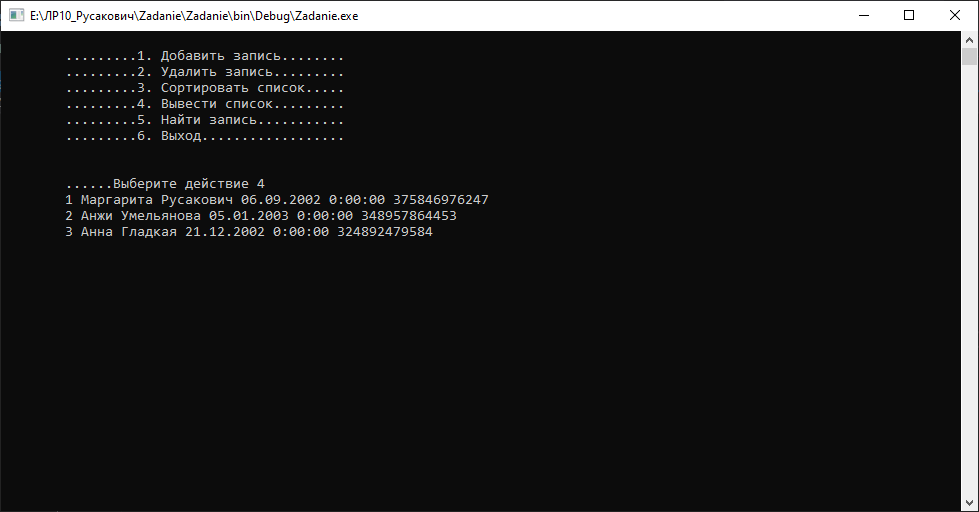


Рисунок 10 – Вывод списка

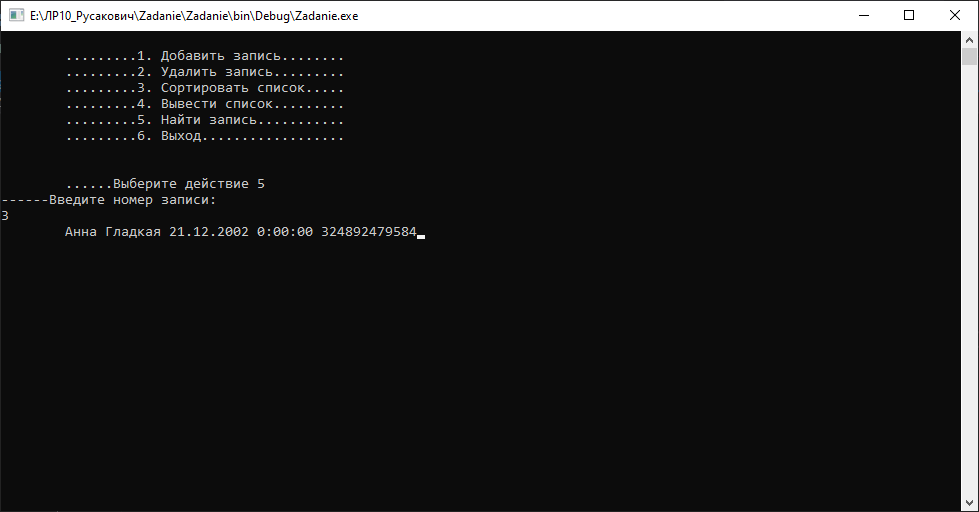


Рисунок 9 – Нахождение записи

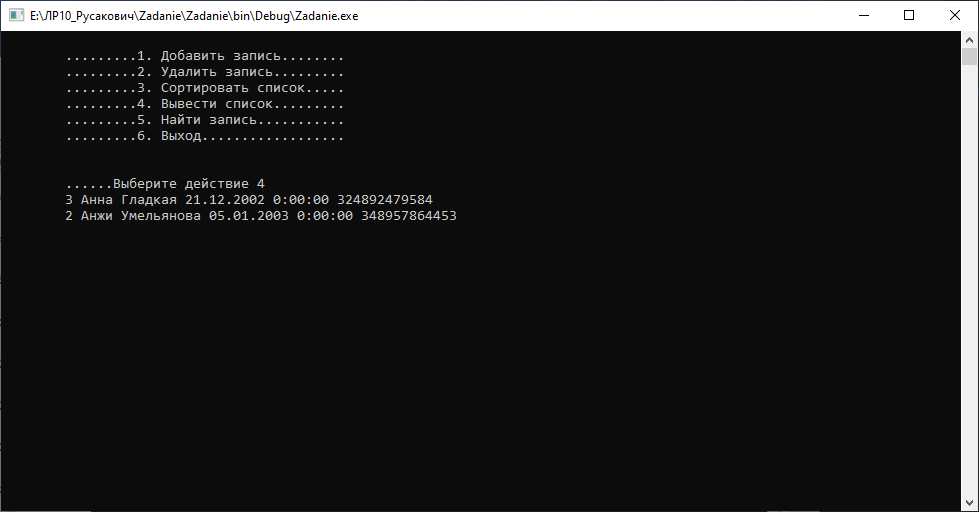


Рисунок 9 – Вывод списка

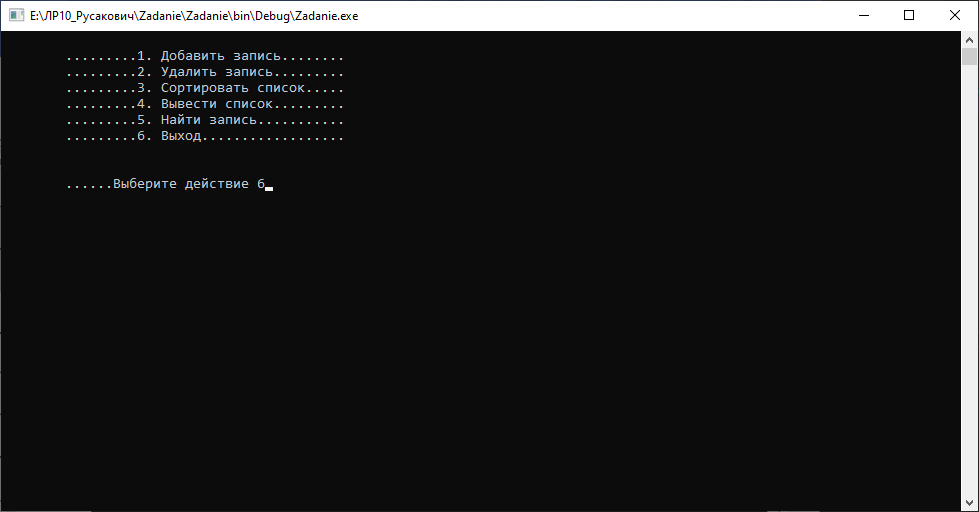


Рисунок 9 – Вывод

**Вывод:** В результате выполнения лабораторной работы были получены практические навыки по использованию механизма наследования в языке программирования C#.