using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab\_4

{

class Program

{

class myClass

{

private int a = 5;

private int b = 7;

public int C

{

get

{

return a \* b;

}

set

{

}

}

}

static void Main(string[] args)

{

myClass A = new myClass();

Console.WriteLine("Результат операции = "+A.C);

Console.ReadKey();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab\_4

{

class Program

{

class A

{

private int a;

private int b;

public A() { a = 7; b = 5; }

public A(int A, int B) { a = A; b = B; }

public int C1

{

get

{

return a \* b;

}

set

{

}

}

}

class B : A

{

private int d;

public B(int D) { d = D; }

public B(int A, int B, int D):base (A, B) { d = D; }

public int C2

{

get

{

return d\*C1/ 2;

}

set

{

// d = value;

}

}

}

static void Main(string[] args)

{

A a1 = new A();

Console.WriteLine("Результат С1 = " + a1.C1);

A a2 = new A(3, 14);

Console.WriteLine("Результат С1 с новыми значениями = " + a2.C1);

B b1 = new B(4);

Console.WriteLine("Результат С2 = " + b1.C2);

B b2 = new B(1, 2, 3);

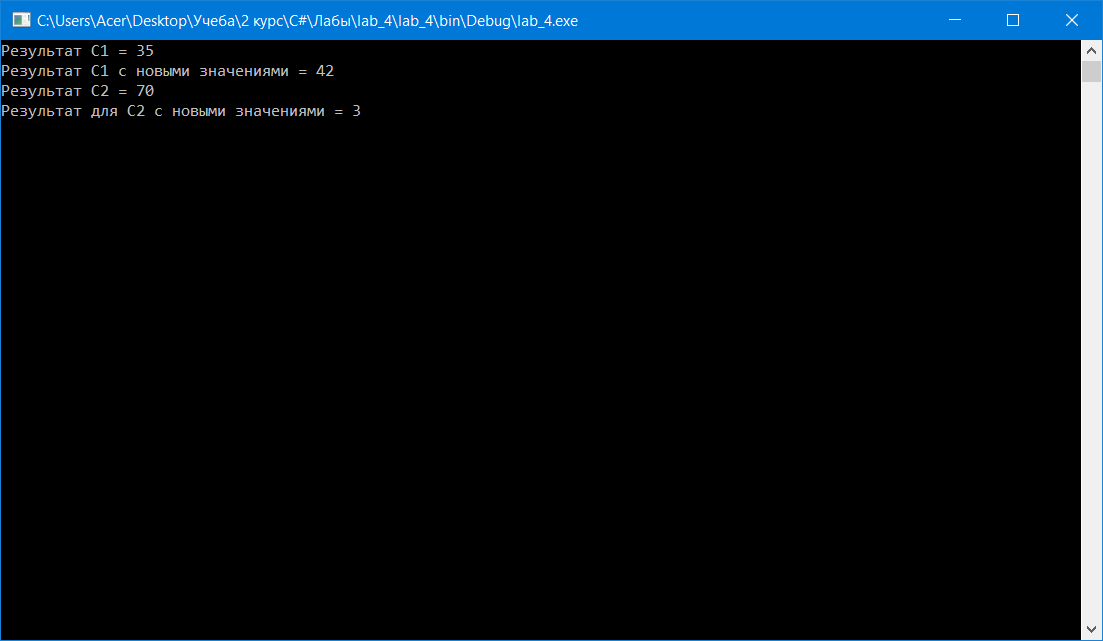
Console.WriteLine("Результат для С2 с новыми значениями = " + b2.C2); ;

Console.ReadKey();

}

}

}



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab\_4

{

class Program

{

class ship

{

private string name;

private string use;

private int weight;

private double power;

private string fuel;

public ship() { name = "first ship"; use = "перевозки"; weight = 54000; power = 4.5; fuel = "бензин"; }

public ship(string A, string B, int C, double D, string E) { name = A; use = B; weight = C; power = D; fuel = E; }

public void Input()

{

Console.WriteLine("Введите название");

name = Console.ReadLine();

Console.WriteLine("Введите предназначение");

use = Console.ReadLine();

Console.WriteLine("Введите тоннаж");

weight = int.Parse(Console.ReadLine());

Console.WriteLine("Введите мощность");

power = double.Parse(Console.ReadLine());

Console.WriteLine("Введите название топлива");

fuel = Console.ReadLine();

}

public void Show()

{

Console.WriteLine("Название - "+name + " " +" для "+ use + " " +"тоннаж - "+ weight + " "+"мощность"+ power + "тип топлива - "+" " + fuel);

}

}

class avia : ship

{

private string type;

private int count;

public avia() { type = "исcтребитель"; count = 120; }

public avia(string F, int T) { type = F;count = T; }

public avia(string A, string B, int C, double D, string E, string F, int T) : base (A, B, C, D, E) { type = F; count = T; }

public void InputAvia()

{

Input();

Console.WriteLine("Введите тип самолётов");

type = Console.ReadLine();

Console.WriteLine("Введите количество самолётов");

count = int.Parse(Console.ReadLine());

}

public void ShowAvia()

{

Show();

Console.WriteLine(" " + type + " " + count+"шт");

}

}

class rocket : ship

{

private string type;

private int count;

public rocket() { type = "ракета"; count = 20; }

public rocket(string M, int N) { type = M;count = N; }

public rocket(string A, string B, int C, double D, string E, string M, int N) :base(A, B, C, D, E) { type = M; count = N; }

public void InputRocket()

{

Input();

Console.WriteLine("Введите тип ракет");

type = Console.ReadLine();

Console.WriteLine("Введите количество ракет");

count = int.Parse(Console.ReadLine());

}

public void ShowRocket()

{

Show();

Console.WriteLine(" " + type + " " + count+"шт");

}

}

static void Main(string[] args)

{

ship a1 = new ship();

a1.Show();

a1.Input();

a1.Show();

avia a2 = new avia();

a2.ShowAvia();

a2.InputAvia();

a2.ShowAvia();

rocket a3 = new rocket();

a3.ShowRocket();

a3.InputRocket();

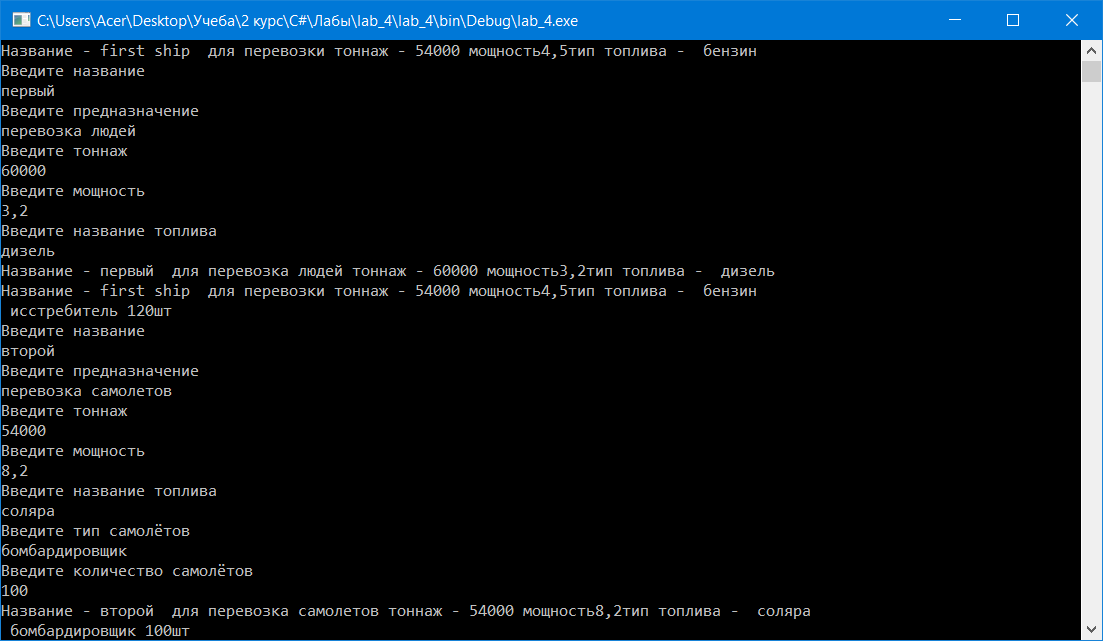
a3.ShowRocket();

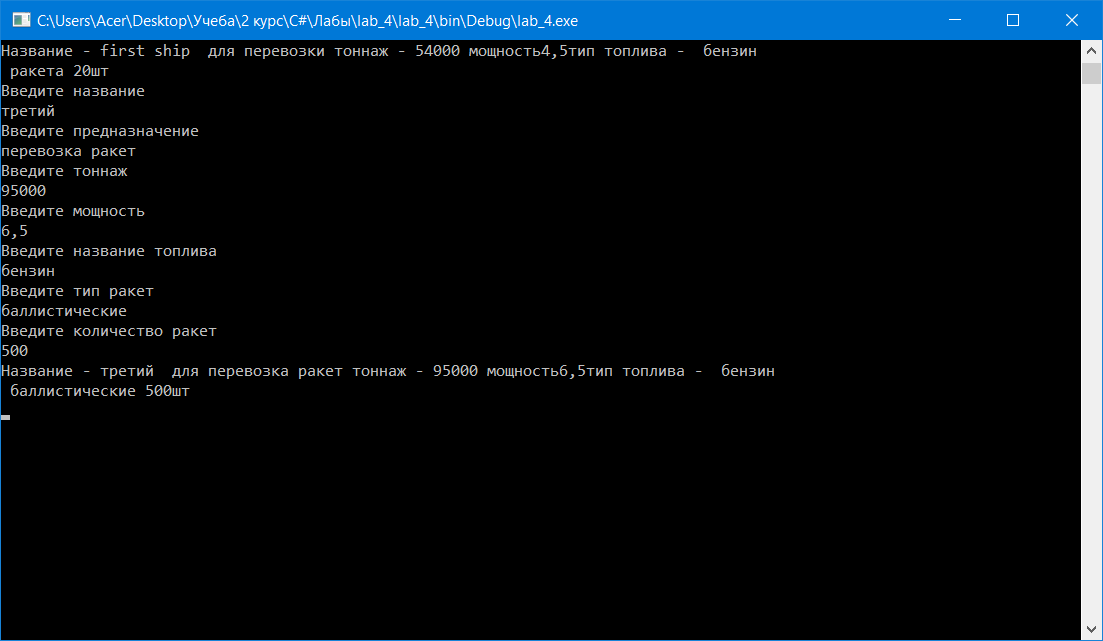
Console.ReadKey();

}

}

}





using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab\_4

{

class Program

{

class sclad

{

private string name;

private string producer;

private string date;

private float price;

private float dimensions;

public void Input()

{

Console.WriteLine("Введите название продукции:");

name = Console.ReadLine();

Console.WriteLine("Введите производителя:");

producer = Console.ReadLine();

Console.WriteLine("Дата поступления на склад:");

date = Console.ReadLine();

Console.WriteLine("Цена товара составляет:");

price = float.Parse(Console.ReadLine());

Console.WriteLine("Введите габариты товара:");

dimensions = float.Parse(Console.ReadLine());

}

public void Show()

{

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine("Название продукции - " + name + ";");

Console.WriteLine("Производитель - " + producer + ";");

Console.WriteLine("Дата поступления на склад - " + date + ";");

Console.WriteLine("Цена товара - " + price + ";");

Console.WriteLine("Габариты товара - " + dimensions + ";");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

public void Search(ref string dez\_date, ref float dez\_price, ref float dez\_dimensions)

{

if ((string.Equals(date, dez\_date)) && price == dez\_price && dimensions == dez\_dimensions)

Show();

}

public void SortByPrice(int n, sclad[] A)

{

float[] prices = new float[n];

for (int i = 0; i != n; i++)

prices[i] = A[i].price;

Array.Sort(prices, A, 0, n);

}

}

static void Main(string[] args)

{

int n;

string dez\_date;

float dez\_price, dez\_dimensions;

Console.WriteLine("Введите количество продукции на складе:");

n = int.Parse(Console.ReadLine());

sclad[] A = new sclad[n];

for (int i = 0; i < n; i++)

{

A[i] = new sclad();

A[i].Input();

A[i].Show();

}

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Поиск товара");

Console.WriteLine("Введите необходимую дату для поиска:");

dez\_date = Console.ReadLine();

Console.WriteLine("Введите необходимую цену для поиска:");

dez\_price = float.Parse(Console.ReadLine());

Console.WriteLine("Введите необходимие габариты для поиска:");

dez\_dimensions = float.Parse(Console.ReadLine());

for (int i = 0; i < n; i++)

{

A[i].Search(ref dez\_date, ref dez\_price, ref dez\_dimensions);

}

Console.WriteLine("Сортировка товара по цене");

for (int i = 0; i < n; i++)

{

A[i].SortByPrice(n, A);

A[i].Show();

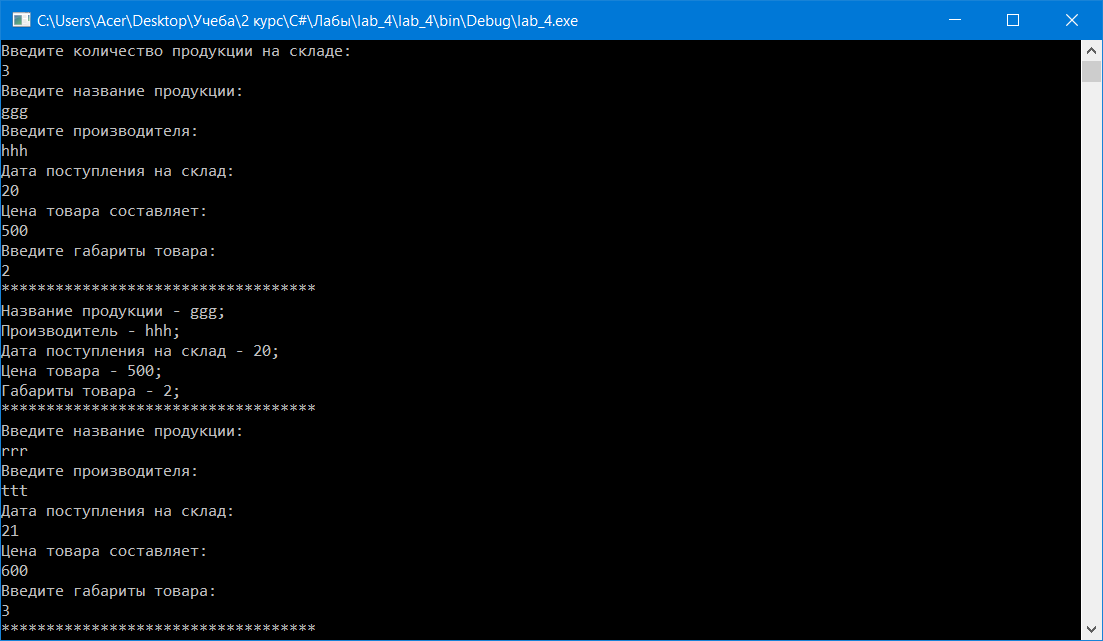
}

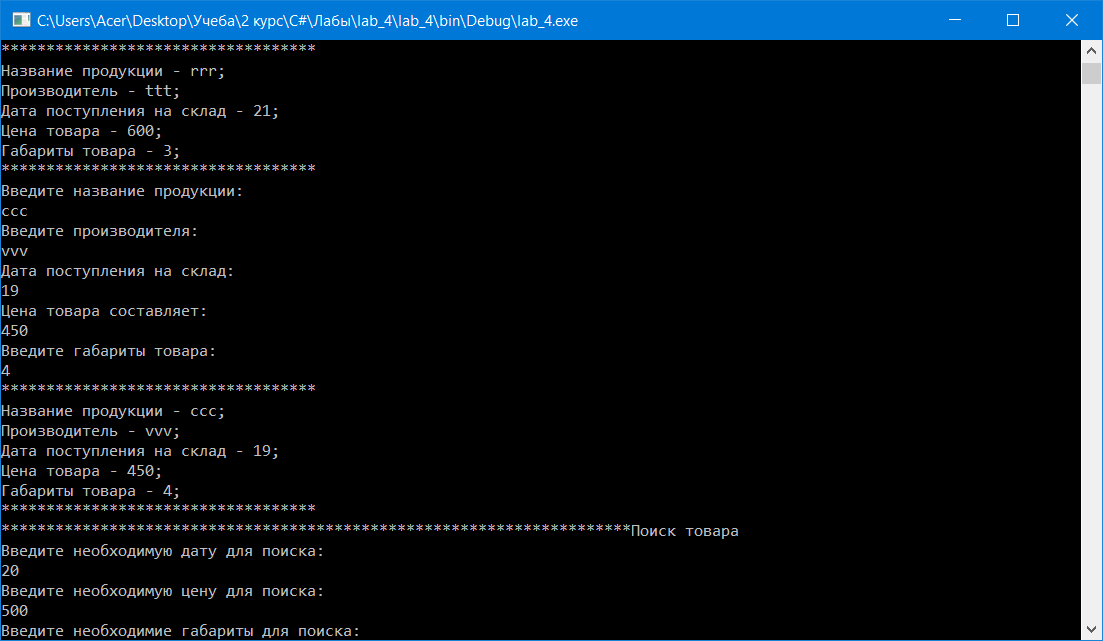
Console.ReadKey();

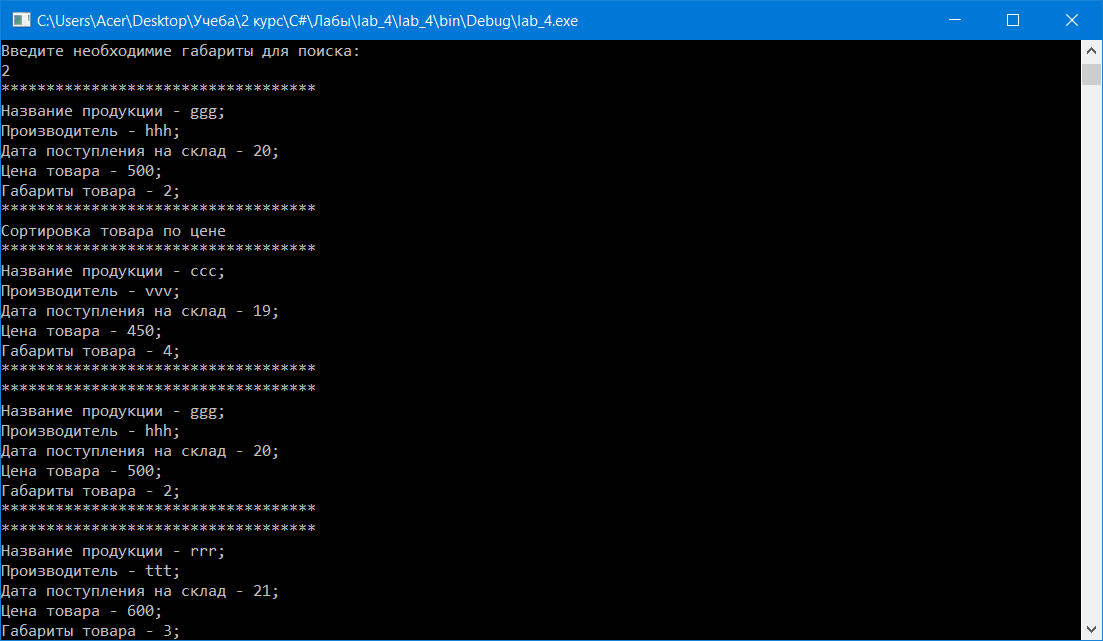
}

}

}







using System;

namespace Student

{

class Program

{

class Student

{

private string name;

private string surname;

private string group;

private float avg\_mark;

private string budget;

public void Input()

{

Console.WriteLine("Enter name of student:");

name = Console.ReadLine();

Console.WriteLine("Enter surname of student:");

surname = Console.ReadLine();

Console.WriteLine("Enter group of student:");

group = Console.ReadLine();

Console.WriteLine("Enter avarage mark of student:");

avg\_mark = float.Parse(Console.ReadLine());

Console.WriteLine("Is student on a budget? (Yes/No):");

budget = Console.ReadLine();

}

public void Output()

{

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine("Name of student - " + name);

Console.WriteLine("Surname - " + surname);

Console.WriteLine("Group - " + group);

Console.WriteLine("Average mark - " + avg\_mark);

Console.WriteLine("Budget? - " + budget);

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

public void Search(ref string s\_surname, ref string s\_group)

{

if ((string.Equals(surname, s\_surname)) && (string.Equals(group, s\_group)) )

Output();

}

public void SortMark(int n, Student[] A)

{

float[] mark = new float[n];

for (int i = 0; i != n; i++)

mark[i] = A[i].avg\_mark;

Array.Sort(mark, A, 0, n);

}

}

static void Main(string[] args)

{

int n;

string s\_surname, s\_group;

try

{

Console.WriteLine("Enter amount of students -");

n = int.Parse(Console.ReadLine());

Student[] A = new Student[n];

for (int i = 0; i < n; i++)

{

A[i] = new Student();

A[i].Input();

A[i].Output();

}

Console.WriteLine("\_\_\_\_\_\_SEARCH\_\_\_\_\_\_");

Console.WriteLine("Enter surname for search -");

s\_surname = Console.ReadLine();

Console.WriteLine("Enter group for search -");

s\_group = Console.ReadLine();

for (int i = 0; i < n; i++)

{

A[i].Search(ref s\_surname, ref s\_group);

}

Console.WriteLine("\_\_\_\_\_\_SORTING\_BY\_MARK\_\_\_\_\_\_");

for (int i = 0; i < n; i++)

{

A[i].SortMark(n, A);

A[i].Output();

}

}

catch (IndexOutOfRangeException)

{

Console.WriteLine("Exception: Выход за грани возможного диапазона.");

}

catch (FormatException)

{

Console.WriteLine("Exception: Недопустимый формат данных.");

}

catch (OutOfMemoryException)

{

Console.WriteLine("Exception: Недостаточно памяти для создания нового обьекта.");

}

Console.ReadKey();

}

}

}