# Главное управление образования Гродненского облисполкома

# Учреждение образования

«Гродненский государственный политехнический колледж»

**О Т Ч Е Т**

по учебной практике по программированию

учащегося 3 курса ПЗТ-33 группы

специальность 2 – 40 01 01 «Программное обеспечение информационных

технологий» место прохождения практики

Выполнил *(подпись)* Я.А. Гончаревич

# Руководитель практики

от предприятия *(подпись) М.П.* А.В. Базукевич

# Руководитель практики

от колледжа В.Д. Орехво

2021

Содержание отчета по практике

1. Описание структуры предприятия, где проходила практика
2. Отчеты по выполненным работам: каждый включает постановку задачи, листинг решения, примеры выполнения.

***Лабораторная работа №1***

***Задание 1***. Для создания нового проекта выполните следующие действия:

1. Создайте новый проект**: File ► New Project**, либо **Сreate Project** в окне **Start Page**

2. В окне **New project** в левой части выберите *Visual C# Projects*, в правой – пункт *Console Application*

3. В поле ***Name*** введите имя проекта, в поле ***Location*** – место его сохранения на диске

4. Ознакомьтесь с основными окнами среды.

5. Рассмотрите каждую строку заготовки программы.

**6.** Наберите приведенный пример. Вставьте свои значения соответствующих типов в пропущенных местах операторов.

Листинг 1.

using System;

namespace ConsoleApp2

{

class Program

{

static void Main(string[] args)

{

int x = 2;

double y = 4;

double z = 12;

Console.WriteLine("x:"+x);

Console.WriteLine("y:"+y);

Console.WriteLine("z:"+z);

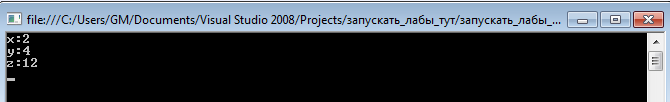
string wait = Console.ReadLine();

}

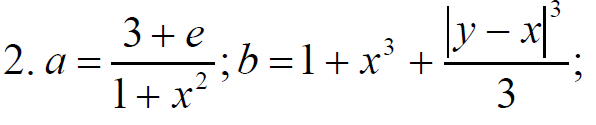
}

}

Пример выполнения.



***Задание 2.*** Создайте новое консольное приложение для решения задачи. Введите вещественные числа x, у, z из области допустимых значений исходных данных. Для преобразования к числовой форме используйте класс Convert и метод Parse. Вычислите а, b. Результаты выведите на экран с использованием формата и шаблонов



Листинг 2.

using System;

namespace ConsoleApp2

{

class Program

{

static void Main(string[] args)

{

Console.Write("x=");

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

Console.Write("y=");

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

Console.Write("z=");

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

double a = (3 + Math.E) / (1 + x \* x);

double b = 1 + x \* x \* x + (Math.Pow(Math.Abs(y - x), 3) / 3);

Console.WriteLine("a = {0,5:N}",a);

Console.WriteLine("b = {0,5:N}",b);

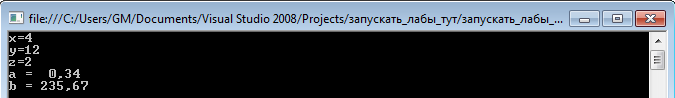
string wait = Console.ReadLine();

}

}

}

Пример выполнения.



***Лабораторная работа №2***

***Задание 1.*** Создайте проект для решения задачи: На экран выводить исходные данные и результаты. В работе использовать только стандартные типы: числовые, символьный и булевский.

2. Введите целое число. Если оно больше 255, вычислите корень квадратный из него, если меньше и соответствует отображаемым символам кодовой таблицы, выведите символ и его код, и комментарий.

Листинг 1.

using System;

namespace ConsoleApp2

{

class Program

{

static void Main(string[] args)

{

Console.Write("x:");

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

if (x > 255)

{

Console.WriteLine("Квадратный корень числа х: {0}", Math.Sqrt(x));

}

else {

if (x >= 0)

{

Console.WriteLine("Символ - {0}, код символа - {1}", Convert.ToChar(x), x);

}

else {

Console.WriteLine("Данному коду({0}) нет соответствующего символа", x);

}

}

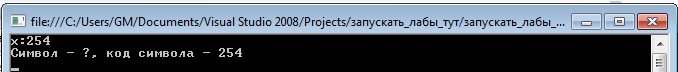
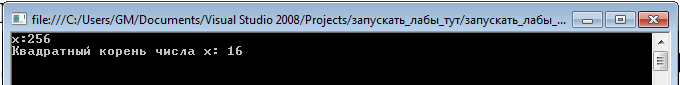
string wait = Console.ReadLine();

}

}

}

Пример выполнения.



***Задание 2.*** Напишите функции в виде методов. Напишите тестирующую программу с выдачей результатов на экран.

2. Напишите программу последовательного ввода чисел из диапазона 0…255. Запрещается последовательно вводить два числа, разность между которыми меньше 7. Программа заканчивает работу после обнаружения первой ошибки.

Листинг 2.

using System;

namespace ConsoleApp2

{

class Program

{

static void Main(string[] args)

{

string trbl="";

int pred;

Console.Write("Введите х:");

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

if (x > 255 | x < 0)

{

trbl = "Ошибка: число х выходит из диапозона(0..255)";

}

else

{

pred = x;

while (trbl == "")

{

Console.Write("Введите х:");

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

if (x > 255 | x < 0)

{

trbl = "Ошибка: число х выходит из диапозона(0..255)";

break;

}

if (x - pred < 7)

{

trbl = "Ошибка: разность введенного числа и предыдущего меньше 7";

break;

}

pred = x;

}

}

Console.Write(trbl);

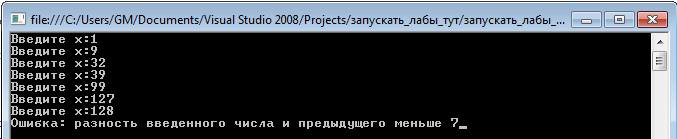
string wait = Console.ReadLine();

}

}

}

Пример выполнения.



***Лабораторная работа №3***

***Задание 1.*** Создайте проект, в котором опишите класс для решения задачи Вашего варианта.

Разрабатываемый класс должен содержать следующие элементы: скрытые и открытые поля, конструкторы без параметров и с параметрами (имена некоторых полей должны совпадать с идентификаторами параметров), методы и свойства. Методы и свойства должны обеспечивать непротиворечивый и удобный интерфейс класса.

В программе должна выполняться проверка всех разработанных элементов класса, вывод состояния объекта.

2. Описать класс, представляющий линейное уравнение вида *aх + b* = 0. Описать метод, вычисляющий решение этого уравнения.

Листинг 1.

using System;

namespace ConsoleApp2

{

class laba3v2

{

public int a;

int b;

int x;

public int mya

{

get { return a; }

set { a = value; }

}

public laba3v2(int a, int b)

{

this.a = a;

this.b = b;

}

public laba3v2()

{

this.a = 1;

this.b = 2;

}

public void reshenie()

{

x = (-1 \* b) / a;

Console.WriteLine("x в уравнении {0}x+{1}=0 равен {2}", a, b, x);

}

}

class Program

{

static void Main(string[] args)

{

laba3v2 zadanie1 = new laba3v2(3, 15);

zadanie1.reshenie();

zadanie1 = new laba3v2();

zadanie1.reshenie();

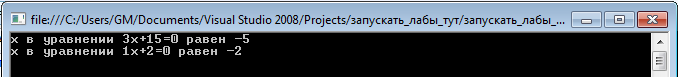
string wait = Console.ReadLine();

}

}

}

Пример выполнения.



***Задание 2***. Включите в проект ***Задания 1*** обработку исключений.

Листинг 2.

using System;

namespace ConsoleApp2

{

class laba3v2{

public int a;

int b;

int x;

public int mya{

get { return a; }

set{ if(value>=0) a = value;}

}

public laba3v2(int a,int b) {

this.a = a;

this.b = b;

}

public laba3v2() {

this.a = 1;

this.b = 2;

}

public void reshenie() {

x =(-1\*b)/a ;

if (b>=0){

Console.WriteLine("x в уравнении {0}x+{1}=0 равен {2}",a,b,x);

}

else {

Console.WriteLine("x в уравнении {0}x{1}=0 равен {2}",a,b,x);

}

}

}

class Program

{

static void Main(string[] args)

{

laba3v2 zadanie1 = new laba3v2(3,-15);

zadanie1.reshenie();

zadanie1 = new laba3v2();

zadanie1.reshenie();

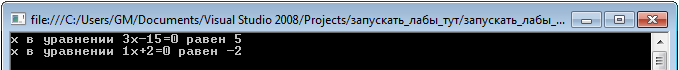
string wait = Console.ReadLine();

}

}

}

Пример выполнения.



***Лабораторная работа №4***

***Задание 1.*** Создайте проект, в котором опишите класс для решения задачи Вашего варианта. Каждый разрабатываемый класс должен содержать следующие элементы: скрытые и открытые поля, конструкторы с параметрами и без параметров, методы, свойства, индексаторы.

Класс должен реализовывать следующие операции над массивами:

* задание произвольной размерности массива при создании объекта;
* доступ к элементу по индексам с контролем выхода за пределы массива;
* вывод на экран элемента массива по заданному индексу и всего массива.

При возникновении ошибок должны выбрасываться исключения.

В программе должна выполняться проверка всех разработанных элементов класса.

2. Описать класс для работы с двумерным массивом символов, состоящих из одних цифр. Обеспечить следующие возможности: рассматривая символы как числа, определить сумму четных и нечетных цифр в каждой строке.

Листинг 1.

using System;

namespace ConsoleApp2

{

class Lab\_Array\_v2{

int a;

int b;

string[,] arr;

public string otvet ="";

void zapolnenie(int a, int b) {

this.a = a;

this.b = b;

string chek;

this.arr = new string[a, b];

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

{

for (int j = 0; j < b; j++)

{

Console.Write("Элемент [{0},{1}] - ", i, j);

chek = Console.ReadLine();

for (int l = 0; l < chek.Length; l++)

{

if (char.IsDigit(chek[l]) == false)

{

chek = "";

break;

}

}

if (chek != "")

{

arr[i, j] = chek;

}

else

{

while (chek == "")

{

Console.WriteLine("Введите целое число!");

Console.Write("Элемент [{0},{1}] - ", i, j);

chek = Console.ReadLine();

for (int l = 0; l < chek.Length; l++)

{

if (char.IsDigit(chek[l]) == false)

{

chek = "";

break;

}

}

if (chek != "")

{

arr[i, j] = chek;

}

}

}

}

}

}

public Lab\_Array\_v2(int a,int b) {

zapolnenie(a,b);

}

public Lab\_Array\_v2()

{

this.a = 3;

this.b = 3;

zapolnenie(a, b);

}

public void reshenie() {

int chet;

int ne\_chet;

for (int i = 0; i < a; i++) {

chet = 0;

ne\_chet = 0;

for (int j = 0; j < a; j++) {

for (int l = 0; l < arr[i, j].Length;l++ )

{

if (int.Parse(arr[i, j][l].ToString()) % 2 == 0)

{

chet += int.Parse(arr[i, j][l].ToString());

}

else

{

ne\_chet += int.Parse(arr[i, j][l].ToString());

}

}

}

otvet = otvet + "Строка №"+(i+1)+": сумма четных цифр: "+chet+"; сумма нечетных цифр: "+ne\_chet+".\n";

}

Console.WriteLine(otvet);

}

public void print\_arr() {

for (int i = 0; i < a; i++) {

Console.Write("[");

for (int j = 0; j < a; j++) {

Console.Write(" "+arr[i, j]);

}

Console.WriteLine(" ]");

}

}

string this[int i, int j] {

get {

if (i >= 0 && i < a) {

if (j >= 0 && j < b)

{

return arr[i, j];

}

else {

Console.WriteLine("Введенный индекс выходит за границы массива");

return "0";

}

}

else { Console.WriteLine("Введенный индекс выходит за границы массива");

return "0";

}

}

set {

if (i >= 0 && i < a)

{

if (j >= 0 && j < b)

{

string chek;

chek = value;

for (int l = 0; l < chek.Length; l++)

{

if (char.IsDigit(chek[l]) == false)

{

chek = "";

break;

}

}

if (chek == "")

{

Console.WriteLine("Введенное значение некорректно");

}

arr[i, j] = chek;

}

else

{

Console.WriteLine("Введенный индекс выходит за границы массива");

}

}

else {

Console.WriteLine("Введенный индекс выходит за границы массива");

}

}

}

}

class Program

{

static void Main(string[] args)

{

Lab\_Array\_v2 zadanie1 = new Lab\_Array\_v2(3, 3);

zadanie1.print\_arr();

zadanie1.reshenie();

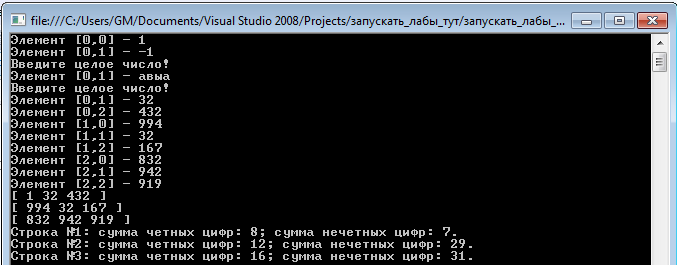
string wait = Console.ReadLine();

}

}

}

Пример выполнения.



***Лабораторная работа №5***

***Задание 1.*** Создайте проект, в котором опишите класс для решения задачи Вашего варианта.

Каждый разрабатываемый класс должен, содержать следующие элементы: скрытые и открытые поля, конструкторы (один из них должен передавать параметром массив), перегруженные операции.

В программе должна выполняться проверка всех разработанных элементов класса.

2. Описать класс для работы с одномерными массивами чисел. Класс должен реализовывать возможность: выполнения для массивов комбинированных операций присваивания (+=, -=).

Листинг 1.

using System;

namespace ConsoleApp2

{

class Lab\_5\_v2

{

public int[] arr;

public Lab\_5\_v2(int[] a)

{

this.arr = a;

}

public Lab\_5\_v2()

{

this.arr = new int[10];

for (int i = 0; i < 10; i++)

{

this.arr[i] = i;

}

}

public static Lab\_5\_v2 operator +(Lab\_5\_v2 op1, Lab\_5\_v2 op2)

{

int[] largest = (op1.arr.Length > op2.arr.Length) ? op1.arr : op2.arr;

int[] shortest = (op1.arr.Length > op2.arr.Length) ? op2.arr : op1.arr;

if (op1.arr.Length > op2.arr.Length)

{

Lab\_5\_v2 r = new Lab\_5\_v2(largest);

for (int i = 0; i < largest.Length; i++)

{

if (i >= shortest.Length)

{

r.arr[i] = op1.arr[i];

}

else

{

r.arr[i] = op1.arr[i] + op2.arr[i];

}

}

return r;

}

else

{

Lab\_5\_v2 r = new Lab\_5\_v2(shortest);

for (int i = 0; i < shortest.Length; i++)

{

r.arr[i] = op1.arr[i] + op2.arr[i];

}

return r;

}

}

public static Lab\_5\_v2 operator -(Lab\_5\_v2 op1, Lab\_5\_v2 op2)

{

int[] largest = (op1.arr.Length > op2.arr.Length) ? op1.arr : op2.arr;

int[] shortest = (op1.arr.Length > op2.arr.Length) ? op2.arr : op1.arr;

if (op1.arr.Length > op2.arr.Length)

{

Lab\_5\_v2 r = new Lab\_5\_v2(largest);

for (int i = 0; i < largest.Length; i++)

{

if (i >= shortest.Length)

{

r.arr[i] = op1.arr[i];

}

else

{

r.arr[i] = op1.arr[i] - op2.arr[i];

}

}

return r;

}

else

{

Lab\_5\_v2 r = new Lab\_5\_v2(shortest);

for (int i = 0; i < shortest.Length; i++)

{

r.arr[i] = op1.arr[i] - op2.arr[i];

}

return r;

}

}

public void show()

{

for (int i = 0; i < arr.Length; i++)

{

Console.Write("{0} ", arr[i]);

}

}

}

class Program

{

static void Main(string[] args)

{

Lab\_5\_v2 a = new Lab\_5\_v2();

int[] array = { 0, 5, 68, 74, 45 };

Lab\_5\_v2 b = new Lab\_5\_v2(array);

Console.WriteLine("Массив а до вычитания:");

a.show();

Console.WriteLine();

Console.WriteLine("Массив b до сложения:");

b.show();

Console.WriteLine();

Console.WriteLine("Массив a после вычитания массива b:");

a -= b;

a.show();

Console.WriteLine();

Console.WriteLine("Массив b после сложения с массивом а:");

b += a;

b.show();

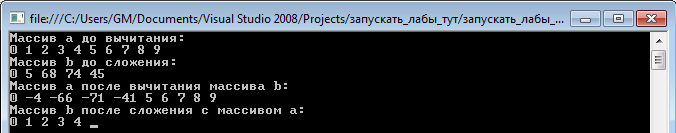
string wait = Console.ReadLine();

}

}

}

Пример выполнения.



***Лабораторная работа №6***

***Задание 1.*** Составить программу с одним родительским классом и потомком. Все поля должны быть закрытыми. Базовый класс должен содержать конструкторы с параметрами, методы доступа к закрытым полям, вывод полей и указанный в таблице метод. Производный класс содержит дополнения и изменения, организовать вывод новых полей потомка, при этом имена методов совпадают с именами методов базового класса. Составить тестирующую программу с выдачей результатов. Создать объекты базового и производного типов. В программе должна выполняться проверка всех разработанных элементов класса.

2. Базовый класс: Сотрудник (поля: имя, р – минимальная зарплата)

Метод: Доход k \*p, где k – повышающий коэффициент.

Потомок: Инженер (поле количество разработанных проектов - n)

Изменения в потомках: Доход инженера увеличить в n/10 раз.

Листинг 1.

using System;

namespace ConsoleApp2

{

class Sotrudnik{

protected string name;

protected double p;

double k;

public Sotrudnik(string name, double p) {

this.name = name;

this.p = p;

this.k=1;

}

public string Name {

get{return name;}

set { name = value; }

}

public double P {

get{return p;}

set { p = value; }

}

public double dohod () {

return k\*p;

}

public void show() {

Console.Write("Имя:{0}; Минимальная ЗП:{1}; ",name,p);

}

}

class Engineer:Sotrudnik{

int n;

public Engineer(string name, double p, int n)

: base(name, p)

{

this.n = n;

this.p = base.p\*(n / 10);

}

public void show() {

base.show();

Console.Write("Кол-во проектов:{0} ",n);

}

}

class Program

{

static void Main(string[] args)

{

Sotrudnik rabochii1 = new Sotrudnik("Имя 1",700);

Engineer rabochii2 = new Engineer("Имя 2", 700,20);

rabochii1.show();

Console.WriteLine();

rabochii2.show();

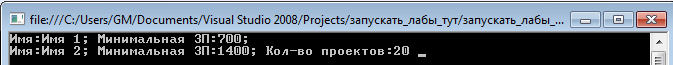
string wait = Console.ReadLine();

}

}

}

Пример выполнения.



***Лабораторная работа №7***

***Задание 1.*** Составить программу с одним родительским классом и двумя потомками. Потомки должны содержать виртуальные функции. Создать виртуальную функцию выдачи результатов расчета методов на экран монитора с указанием названий и полей и их значений соответствующего объекта. Составить тестирующую программу с выдачей протокола на экран монитора. При этом создать объекты базового и производных типов, используя полиморфный контейнер - массив ссылок базового класса на объекты базового и производных классов (количество объектов >=5).

|  |  |  |  |
| --- | --- | --- | --- |
| 2. | Транспортное средство (поле - название) | Самолет (высота - h, скорость - v)  Корабль (количество пассажиров k, порт приписки) | Стоимость транспортного средства  Самолет 100\*h\*v  Корабль 5k2 |

Листинг 1.

using System;

namespace ConsoleApp2

{

class Transport{

public string name;

public Transport(string nam) {

name = nam;

}

virtual public void Print() {

Console.WriteLine("Название - {0}", name);

}

}

class Plane : Transport {

int h;

int v;

public Plane (string name, int h1, int v1) : base(name) {

h = h1;

v = v1;

}

override public void Print() {

Console.WriteLine("Стоимость Самолета \"{0}\" - {1}",name,100\*v\*h);

}

}

class Ship : Transport {

int k;

string port;

public Ship(string name, int k1, string port1):base(name) {

k = k1;

port = port1;

}

override public void Print()

{

Console.WriteLine("Стоимость Корабля \"{0}\" - {1}", name, 5 \* k \* k);

}

}

class Program

{

static void Main(string[] args)

{

const int n = 7;

Transport[] st = new Transport[n];

st[0]= new Plane("plane-231",3,200);

st[1] = new Ship("ship-762",100,"port34");

st[2] = new Plane("plane-542", 2, 300);

st[3] = new Ship("ship-76", 1043, "port51");

st[4] = new Plane("plane-28", 3, 175);

st[5] = new Ship("ship-710", 100, "port754");

st[6] = new Transport("Auto-432");

foreach (Transport elem in st) {

elem.Print();

}

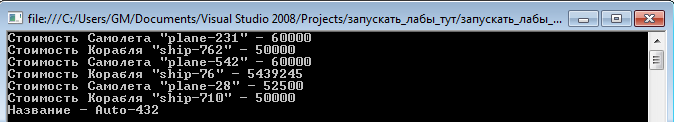
string wait = Console.ReadLine();

}

}

}

Пример выполнения.



***Задание 2.*** Составить программу с абстрактным родительским клас-сом и двумя объектами - потомками. Для этого модифицировать задание 1. Составить тестирующую программу с выдачей протокола на экран мони-тора. В ней нужно реализовать циклический вывод параметров объектов, используя полиморфный контейнер - массив объектов базового класса (ко-личество объектов >=5).

2. Организовать вычисление средней стоимости самолетов и средней стоимости кораблей.

Листинг 2.

using System;

namespace ConsoleApp2

{

abstract class Transport{

public string name;

public string type\_of\_transport;

public int cost;

public Transport(string nam) {

name = nam;

}

abstract public void Print();

}

class Plane : Transport {

int h;

int v;

public Plane (string name, int h1, int v1) : base(name) {

h = h1;

v = v1;

cost = 100 \* v \* h;

type\_of\_transport = "plane";

}

override public void Print() {

Console.WriteLine("Стоимость Самолета \"{0}\" - {1}",name,cost);

}

}

class Ship : Transport {

int k;

string port;

public Ship(string name, int k1, string port1):base(name) {

k = k1;

port = port1;

cost = k\*k\*5;

type\_of\_transport = "ship";

}

override public void Print()

{

Console.WriteLine("Стоимость Корабля \"{0}\" - {1}", name, cost);

}

}

class Program

{

static void Main(string[] args)

{

const int n = 6;

double cost\_plane = 0;

int count\_plane = 0;

double cost\_ship = 0;

int count\_ship = 0;

Transport[] st = new Transport[n];

st[0]= new Plane("plane-231",3,200);

st[1] = new Ship("ship-762",100,"port34");

st[2] = new Plane("plane-542", 2, 300);

st[3] = new Ship("ship-76", 1043, "port51");

st[4] = new Plane("plane-28", 3, 175);

st[5] = new Ship("ship-710", 100, "port754");

foreach (Transport elem in st) {

if (elem.type\_of\_transport == "plane")

{

cost\_plane += elem.cost;

count\_plane++;

}

else

{

if (elem.type\_of\_transport == "ship")

{

cost\_ship += elem.cost;

count\_ship++;

}

}

}

cost\_plane = cost\_plane / count\_plane;

cost\_ship = cost\_ship / count\_ship;

Console.WriteLine("Средняя стоимость всех самолетов - {0}\nСредняя стоимость всех кораблей - {1}",cost\_plane,cost\_ship);

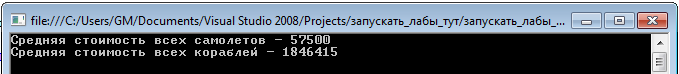
string wait = Console.ReadLine();

}

}

}

Пример выполнения.



***Лабораторная работа №8***

***Задание 1.*** Интерфейсы Ix, Iy, Iz, содержат объявления методов с одной и той же сигнатурой следующим образом

**interface** Ix

{

**void** IxF0(параметр);

**void** IxF1();

}

**interface** Iy

{

**void** F0(параметр);

**void** F1();

}

**interface** Iz

{

**void** F0(параметр);

**void** F1();

}

Эти интерфейсы наследуются в классе TestClass, содержащий член w типа параметр и реализуются так, как задано в варианте. В каждом методе задать вывод результата.

Рассмотреть случай

* неявной реализации интерфейсов
* явной реализации интерфейса Iz

В программе должна выполняться:

* неявная неоднозначная реализация методов интерфейсов Iy и Iz,
* вызов функций с явным приведением к типу интерфейса,
* вызов метода для объекта посредством интерфейсной ссылки.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | параметр | IxF0, IxF1 возвращают | F0 F1 возвращают | |
| Неявная реализация | Явная реализация Iz |
| 2 | **double** | Cos(w) | Exp(w) | 1/Exp(w) |

Листинг 1.

using System;

namespace ConsoleApp2

{

interface Ix

{

void IxF0(double w);

void IxF1();

}

interface Iy

{

void F0(double w);

void F1();

}

interface Iz

{

void F0(double w);

void F1();

}

class TestClass : Ix, Iy, Iz

{

double w;

public TestClass(double ww)

{

w = ww;

}

public TestClass()

{

w = 1;

}

public void IxF0(double ww)

{

w = Math.Cos(ww);

Console.WriteLine("cos({0}) = {1}", ww, w);

}

public void IxF1()

{

Console.Write("cos({0}) =", w);

w = Math.Cos(w);

Console.WriteLine("{0}", w);

}

public void F0(double ww)

{

w = Math.Exp(ww);

Console.WriteLine("Exp({0}) = {1}", ww, w);

}

public void F1()

{

Console.Write("Exp({0}) =", w);

w = Math.Exp(w);

Console.WriteLine("{0}", w);

}

void Iz.F0(double ww)

{

w = 1 / Math.Exp(ww);

Console.WriteLine("1/Exp({0}) = {1}", ww, w);

}

void Iz.F1()

{

Console.Write("1/Exp({0}) =", w);

w = 1 / Math.Exp(w);

Console.WriteLine(" {0}", w);

}

}

class Program

{

static void Main(string[] args)

{

TestClass ob = new TestClass(7);

ob.IxF0(7);

ob.IxF1();

(ob as Iy).F0(12);

(ob as Iz).F0(12);

(ob as Iy).F1();

(ob as Iz).F1();

Iy y\_ob;

y\_ob = ob;

y\_ob.F0(12);

y\_ob.F1();

Iz z\_ob;

z\_ob = ob;

z\_ob.F0(12);

z\_ob.F1();

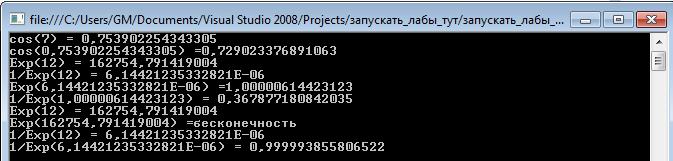
string wait = Console.ReadLine();

}

}

}

Пример выполнения.



***Задание 2.*** Выполнить задания, используя для хранения экземпляров разработанных классов стандартные параметризованные коллекции. Во всех классах реализовать интерфейсы IComparable и IComparer перегрузить операции отношения для реализации сравнения объектов по указанному полю. Результат вывести на экран.

2. Составить список студентов, включающий ФИО, курс, группу, результат забега. Вывести в новый список информацию о студентах, показавших три лучших результата в забеге. Если окажется, что некоторые студенты получили такие же высокие результаты, то добавить их к списку победителей, отсортировать по результатам.

Листинг 2.

using System;

using System.Collections;

using System.IO;

using System.Text;

namespace ConsoleApp2

{

class Student : IComparable

{

public string fio;

public int kurs;

public string grup;

public int rezultat;

public override string ToString()

{

return String.Format("ФИО: {0}; Курс: {1}; Группа: {2}; Результат: {3}.", fio, kurs, grup, rezultat);

}

public int CompareTo(object obj)

{

Student b;

b = (Student)obj;

return rezultat.CompareTo(b.rezultat);

}

}

class SortByRezultat : IComparer

{

int IComparer.Compare(object x, object y)

{

Student st1 = (Student)x;

Student st2 = (Student)y;

if (st1.rezultat < st2.rezultat) return 1;

if (st1.rezultat > st2.rezultat) return -1;

return 0;

}

}

class Program

{

static void Main(string[] args)

{

const int n = 5;

ArrayList students = new ArrayList();

ArrayList champions = new ArrayList();

ArrayList final\_champions = new ArrayList();

Console.WriteLine("Введите данные о студентах:");

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

{

Student a = new Student();

Console.Write("ФИО: ");

a.fio = Console.ReadLine();

Console.Write("Курс: ");

a.kurs = int.Parse(Console.ReadLine());

Console.Write("Группа: ");

a.grup = Console.ReadLine();

Console.Write("Результат: ");

a.rezultat = int.Parse(Console.ReadLine());

students.Add(a);

}

students.Sort(new SortByRezultat());

for (int i = 0; i < 3; i++)

{

champions.Add(students[i]);

}

champions.Sort(new SortByRezultat());

Console.WriteLine("Информация о студентах после сортировки:");

foreach (Student a in students)

{

foreach (Student b in champions)

{

if (a.rezultat >= b.rezultat)

{

final\_champions.Add(a);

break;

}

}

}

final\_champions.Sort(new SortByRezultat());

foreach (Student chmp in final\_champions)

{

Console.WriteLine("{0}", chmp);

}

string wait = Console.ReadLine();

}

}

}

Пример выполнения.

