**9 «ИНТЕРФЕЙСЫ».**

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.

Листинг программы:

using System;

interface Ix

{

void IxF0(double parameter);

void IxF1();

}

interface Iy

{

void F0(double parameter);

void F1();

}

interface Iz

{

void F0(double parameter);

void F1();

}

class TestClass : Ix, Iy, Iz

{

private double w;

void Iz.F0(double parameter)

{

Console.WriteLine($"Iz.F0: {Math.Sqrt(parameter)}");

}

void Iz.F1()

{

Console.WriteLine("Iz.F1");

}

public void IxF0(double parameter)

{

Console.WriteLine($"Ix.IxF0: {Math.Sqrt(parameter)}");

}

public void IxF1()

{

Console.WriteLine("Ix.IxF1");

}

public void F0(double parameter)

{

Console.WriteLine($"Iy.F0: {parameter \* parameter + 5}");

}

public void F1()

{

Console.WriteLine("Iy.F1");

}

}

class Program

{

static void Main()

{

TestClass obj = new TestClass();

obj.F0(4);

((Iz)obj).F0(4);

((Ix)obj).IxF0(16);

((Ix)obj).IxF1();

Iy iyObj = obj;

iyObj.F1();

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Iy.F0: 21  Iz.F0: 2  Ix.IxF0: 4  Ix.IxF1  Iy.F1 |

Анализ результатов:

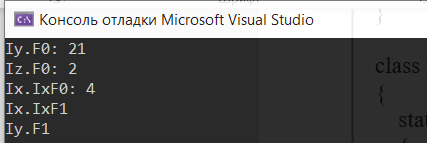


Рисунок 1.1 – Результат работы программы

Задание 2. Описать класс для работы с одномерным массивом строк фиксированной

длины. Обеспечить следующие возможности:

 задание произвольных целых границ индексов при создании объекта;

 обращение к отдельной строке массива по индексу с контролем выхода за

 пределы массива;

 выполнение операций поэлементного сцепления двух массивов с

образованием нового массива;

 вывод на экран элемента массива по заданному индексу и всего массива.

Написать программу, демонстрирующую все разработанные элементы

класса.

Листинг программы:

using System;

class FixedLengthStringArray

{

private string[] array;

private int lowerBound;

private int upperBound;

public FixedLengthStringArray(int lowerBound, int upperBound)

{

if (lowerBound > upperBound)

{

throw new ArgumentException("Lower bound cannot be greater than upper bound.");

}

this.lowerBound = lowerBound;

this.upperBound = upperBound;

array = new string[upperBound - lowerBound + 1];

}

public string this[int index]

{

get

{

if (index < lowerBound || index > upperBound)

{

throw new IndexOutOfRangeException("Index is out of range.");

}

return array[index - lowerBound];

}

set

{

if (index < lowerBound || index > upperBound)

{

throw new IndexOutOfRangeException("Index is out of range.");

}

array[index - lowerBound] = value;

}

}

public FixedLengthStringArray Concatenate(FixedLengthStringArray other)

{

if (this.Length != other.Length)

{

throw new InvalidOperationException("Arrays must have the same length for concatenation.");

}

FixedLengthStringArray resultArray = new FixedLengthStringArray(lowerBound, upperBound);

for (int i = lowerBound; i <= upperBound; i++)

{

resultArray[i] = this[i] + other[i];

}

return resultArray;

}

public void PrintElement(int index)

{

Console.WriteLine($"Element at index {index}: {this[index]}");

}

public void PrintArray()

{

Console.WriteLine("Array:");

for (int i = lowerBound; i <= upperBound; i++)

{

Console.WriteLine($"[{i}]: {this[i]}");

}

}

public int Length => upperBound - lowerBound + 1;

}

class Program

{

static void Main(string[] args)

{

FixedLengthStringArray array1 = new FixedLengthStringArray(0, 4);

FixedLengthStringArray array2 = new FixedLengthStringArray(0, 4);

array1[0] = "Hello";

array1[1] = " ";

array1[2] = "world";

array1[3] = "!";

array1[4] = " ";

array2[0] = "How";

array2[1] = " ";

array2[2] = "are";

array2[3] = " ";

array2[4] = "you?";

array1.PrintArray();

array2.PrintArray();

FixedLengthStringArray concatenatedArray = array1.Concatenate(array2);

concatenatedArray.PrintArray();

array1.PrintElement(2);

array2.PrintElement(3);

}

}

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Array:  [0]: Hello  [1]:  [2]: world  [3]: !  [4]:  Array:  [0]: How  [1]:  [2]: are  [3]:  [4]: you?  Array:  [0]: HelloHow  [1]:  [2]: worldare  [3]: !  [4]: you?  Element at index 2: world  Element at index 3: |

Анализ результатов:

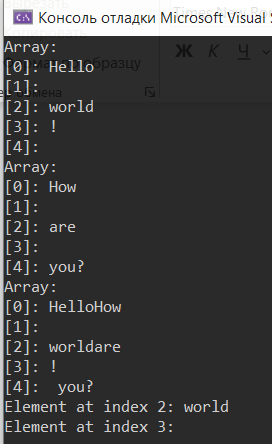


Рисунок 1.2 – Результат работы программы