**9 Интерфейсы**

**Задание №1:** Интерфейсы Ix, Iy, Iz, содержат объявления методов с одной и той же сигнатурой. Эти интерфейсы наследуются в классе TestClass, содержащий член w типа параметр и реализуются так, как задано в варианте. В каждом методе

задать вывод результата.

Листинг программы:

using System;

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

{

private double w;

public TestClass(double w) => this.w = w;

public void IxF0(double w) => Console.WriteLine($"IxF0: w^2 = {w \* w}");

public void IxF1() => Console.WriteLine($"IxF1: √w = {Math.Sqrt(w)}");

public void F0(double w) => Console.WriteLine($"Iy F0: w^2 = {w \* w}");

public void F1() => Console.WriteLine($"Iy F1: √w = {Math.Sqrt(w)}");

void Iz.F0(double w) => Console.WriteLine($"Iz F0: w^2 + 5 = {w \* w + 5}");

void Iz.F1() => Console.WriteLine($"Iz F1: √w = {Math.Sqrt(w)}");

}

class Program

{

static void Main()

{

var test = new TestClass(4.0);

Console.WriteLine("Вызовы через интерфейс Ix:");

test.IxF0(4); test.IxF1();

Console.WriteLine("\nВызовы через интерфейс Iy:");

test.F0(4); test.F1();

Console.WriteLine("\nВызовы через интерфейс Iz:");

((Iz)test).F0(4); ((Iz)test).F1();

Console.WriteLine("\nВызовы через интерфейсную ссылку:");

Ix ix = test;

ix.IxF0(4); ix.IxF1();

}

}

Таблица 9.1. – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| double  w^2  √w  w^2 +5 | Вызовы через интерфейс Ix:  IxF0: w^2 = 16  IxF1: √w = 2  Вызовы через интерфейс Iy:  Iy F0: w^2 = 16  Iy F1: √w = 2  Вызовы через интерфейс Iz:  Iz F0: w^2 + 5 = 21  Iz F1: √w = 2  Вызовы через интерфейсную ссылку:  IxF0: w^2 = 16  IxF1: √w = 2 |

Анализ результатов:

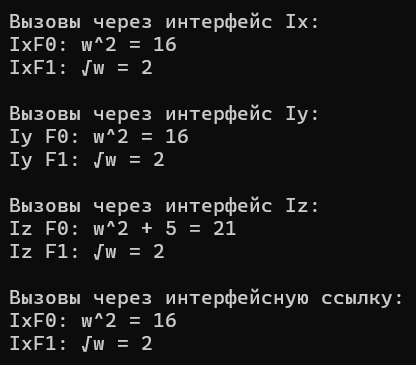


Рисунок 9.1. – Результат работы программы