**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 и 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:

interface Ix{ void IxF0(int w); void IxF1();}

interface Iy{ void F0(int w); void F1();}

interface Iz{ void F0(int w); void F1();}

class TestClass : Ix, Iy, Iz{ public int w;

public void F0(int w) {

Console.WriteLine("Результат неявной реализации метода F0 интерфейса Iy: " + w \* 3); }

public void F1()

{Console.WriteLine("Метод F1 интерфейса Iy выполнен");}

void Iz.F0(int w)

{ Console.WriteLine("Результат явной реализации метода F0 интерфейса Iz: " + (6 + w)); }

void Iz.F1()

{Console.WriteLine("Метод F1 интерфейса Iz выполнен"); }

public void IxF0(int w)

{Console.WriteLine("Результат метода IxF0 интерфейса Ix: " + (7 \* w - 4)); } public void IxF1()

{Console.WriteLine("Метод IxF1 интерфейса Ix выполнен"); }}

class Program

{ static void Main(string[] args){TestClass test = new TestClass();

test.w = 5; ((Iy)test).F0(test.w); ((Iy)test).F1();

((Iz)test).F0(test.w); ((Iz)test).F1(); test.IxF0(test.w); test.IxF1();

Ix ix = test; ix.IxF0(test.w); ix.IxF1(); Iy iy = test; iy.F0(test.w); iy.F1(); Iz iz = test; iz.F0(test.w); iz.F1();}}

Таблица 9.1 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Результат неявной реализации метода F0 интерфейса Iy: 15  Метод F1 интерфейса Iy выполнен  Результат явной реализации метода F0 интерфейса Iz: 11 Метод F1 интерфейса Iz выполнен  Результат метода IxF0 интерфейса Ix: 31  Метод IxF1 интерфейса Ix выполнен  Результат метода IxF0 интерфейса Ix: 31  Метод IxF1 интерфейса Ix выполнен  Результат неявной реализации метода F0 интерфейса Iy: 15  Метод F1 интерфейса Iy выполнен  Результат явной реализации метода F0 интерфейса Iz: 11 Метод F1 интерфейса Iz выполнен |

Источник: собственная разработка

Анализ результатов:

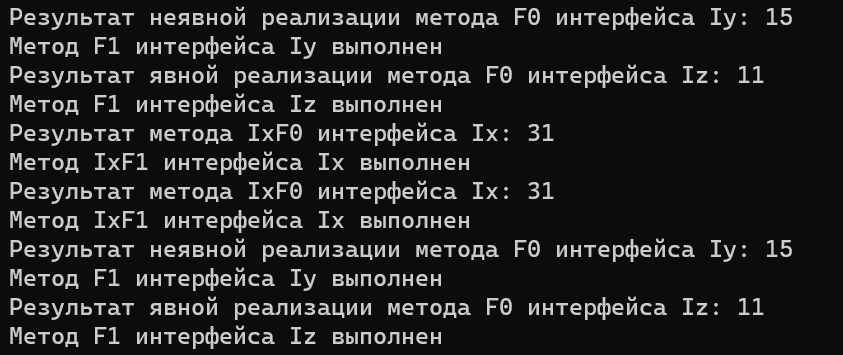


Рисунок 9.1 – Результат работы программы

Источник: собственная разработка

Задание 2. Выполнить задания, используя для хранения экземпляров разработанных классов массивы. Во всех классах реализовать интерфейс IComparable и перегрузить операции отношения для реализации значимой семантики сравнения объектов по какому-либо полю. Описать класс «домашняя библиотека». Предусмотреть возможность работы с произвольным числом книг, поиска книги по какому-либо признаку (по автору, по году издания или категории), добавления книг в библиотеку, удаления книг из нее, доступа к книге по номеру.

Написать программу, демонстрирующую все разработанные элементы класса.

Листинг программы:

class Book : IComparable<Book>{ public string title; public string author;

public int year; public Book(string title, string author, int year)

{ this.title = title; this.author = author; this.year = year; }

public int CompareTo(Book other){ return this.year.CompareTo(other.year); }

}

class HomeLibrary{private Book[] books;

public HomeLibrary() { books = new Book[0]; }

public void AddBook(Book book){Array.Resize(ref books, books.Length + 1);

books[books.Length - 1] = book;}

public void RemoveBook(int index){for (int i = index; i < books.Length - 1; i++)

{books[i] = books[i + 1];}Array.Resize(ref books, books.Length - 1);}

public Book[] FindBooksByYear(int year){

Book[] result = new Book[0]; foreach (Book book in books){

if (book.year == year){Array.Resize(ref result, result.Length + 1);

result[result.Length - 1] = book;}}return result;}

public Book GetBook(int index){return books[index];}}

class Program{static void Main(string[] args) {

HomeLibrary library = new HomeLibrary();

library.AddBook(new Book("Жизнь в коде", "Варченя", 1866));

library.AddBook(new Book("Мастер и Маргарита", "М.А. Булгаков", 1966)); library.AddBook(new Book("11/22/63", "Стивен Кинг", 2011));

library.AddBook(new Book("Унесенные ветром", "М.Митчелл", 1936));

Console.Write("Введите год издания книги, которые хотите найти: ");

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

Book[] booksByYear = library.FindBooksByYear(year);

if (booksByYear.Length == 0){

Console.WriteLine("Книги, изданные в {0} году, не найдены.", year);}

else{ Console.WriteLine("Книги, изданные в {0} году:", year);

foreach (Book book in booksByYear){ Console.WriteLine(book.title + " от " + book.author);}}

Console.Write("Введите номер книги, которую хотите удалить: ");

int bookIndex = int.Parse(Console.ReadLine());library.RemoveBook(bookIndex);

Console.Write("Введите номер книги, которую хотите получить: ");

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

Book bookByNumber = library.GetBook(bookNumber);

Console.WriteLine("Книга под номером {0}: {1}", bookNumber, bookByNumber.title);}}

Таблица 9.2 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 1866, 2, 2 | Введите год издания книги, которые хотите найти: 1866  Книги, изданные в 1866 году:  Жизнь в коде от Варченя  Введите номер книги, которую хотите удалить: 2  Введите номер книги, которую хотите получить: 2  Книга под номером 2: Унесенные ветром |

Источник: собственная разработка

Анализ результатов:

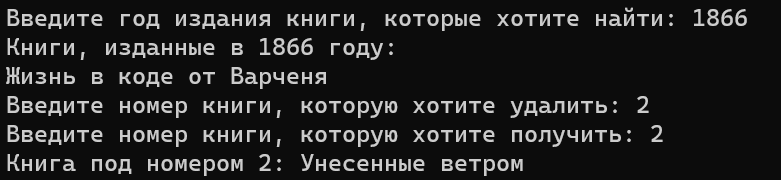


Рисунок 9.2 – Результат работы программы

Источник: собственная разработка