МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Курский государственный университет»

кафедра программного обеспечения и администрирования

информационных систем

Отчёт

по лабораторной работе №6

«Интерфейсы и структурные типы»

по дисциплине

###### «Объектно-ориентированные языки и системы»

|  |  |
| --- | --- |
| Выполнил(а): | студент группы 313  Козявин Максим  Сергеевич |
| Проверил: | к.т.н., доцент  кафедры ПОиАИС  Макаров К.С. |

Курск

2023

**Цель работы:** сформировать понятие о реализации принципа полиморфизма с помощью интерфейсов, умения использовать пользовательские интерфейсы для задания поведения различных классов.

**Индивидуальное задание (вариант 4):**

**Задание 1:**

1. Реализовать неявно интерфейсы в классах 1 и 2, содержащих поле указанного типа. Переменная w обозначает параметр метода или поле. Результат метода со спецификатором void присвоить полю класса.

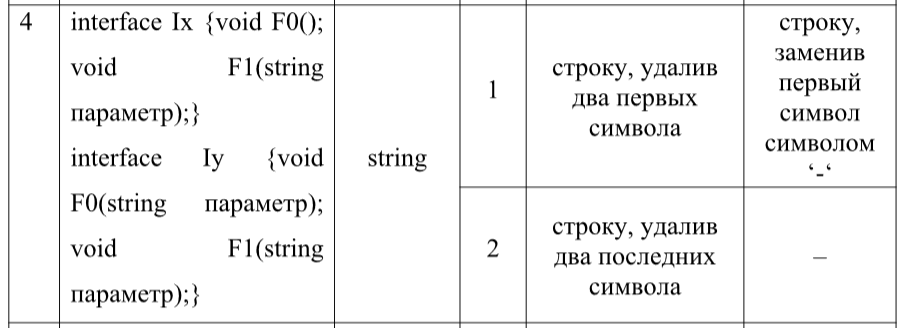
В тестирующей программе должны выполняться:

– неявная неоднозначная реализация методов интерфейсов;

– вызов методов с явным приведением к типу интерфейса;

– вызов методов для объекта посредством интерфейсной ссылки.

1. Модифицировать проект, используя явную реализацию интерфейса для методов с одинаковой сигнатурой в классе 1.



**Задание 2:**

Выполнить задания, используя для хранения экземпляров разработанных классов стандартную коллекцию ArrayList. Во всех классах реализовать интерфейсы IComparable и IComparer. Перегрузить операции отношения для реализации сравнения объектов по указанному полю. Результат вывести на экран. Организовать вывод коллекции.

**Вариант 4**

Составить список студентов группы, включив следующие данные: ФИО, номер группы, результаты сдачи трех экзаменов. Вывести в новый список информацию о студентах, успешно сдавших сессию, отсортировав по номеру группы.

**Основные теоретические положения**

Интерфейс представляет ссылочный тип, который может определять некоторый функционал - набор методов и свойств без реализации. Затем этот функционал реализуют классы и структуры, которые применяют данные интерфейсы.

Интерфейсы могут определять методы, свойства, индексаторы, события и статические поля и константы (начиная с версии C# 8.0).

Начиная с версии C# 8.0 методы и свойства интерфейса могут иметь реализацию.

**Экспериментальные результаты**

Код программы для решения задачи 1:

internal class Program

{

private static void Main(string[] args)

{

Class2 cl2 = new Class2("abcdefg");

cl2.F0();

Console.WriteLine(cl2.Str);

(cl2 as Iy).F1("123456");

Console.WriteLine(cl2.Str);

Ix x\_cl2 = cl2;

x\_cl2.F1("123456");

Console.WriteLine(cl2.Str);

Class1 cl1 = new Class1("abcdefg");

cl1.F0();

Console.WriteLine(cl1.Str);

(cl1 as Iy).F1("123456");

Console.WriteLine(cl1.Str);

Ix x\_cl1 = cl1;

x\_cl1.F1("123456");

Console.WriteLine(cl1.Str);

}

}

class Class1: Iy, Ix {

private string str = "";

public Class1(string str)

{

this.str = str;

}

public string Str

{

get {return str;}

private set {}

}

public void F0() {

if (str.Length > 2)

{

str = str.Substring(2);

}

else

{

str = "";

}

}

public void F0(string p)

{

if (p.Length > 2)

{

str = p.Substring(2);

}

else

{

str = "";

}

}

void Iy.F1(string p)

{

if (str.Length > 0)

{

str = string.Concat("-", str.Substring(1));

}

else

{

str = "-";

}

}

public void F1(string p)

{

if (p.Length > 2)

{

str = p.Substring(2);

}

else

{

str = "";

}

}

}

class Class2: Ix, Iy{

private string str = "";

public Class2(string str)

{

this.str = str;

}

public string Str

{

get {return str;}

private set {}

}

public void F0() {

if (str.Length > 2)

{

str = str.Substring(0, str.Length-2);

}

else

{

str = "";

}

}

public void F0(string p)

{

if (p.Length > 2)

{

str = p.Substring(0, p.Length-2);

}

else

{

str = "";

}

}

public void F1(string p)

{

if (p.Length > 2)

{

str = p.Substring(0, p.Length-2);

}

else

{

str = "";

}

}

}

interface Ix

{

void F0();

void F1(string p);

}

interface Iy

{

void F0(string p);

void F1(string p);

}

Код программы для решения задачи 2:

using System.Collections;

internal class Program

{

private static ArrayList array = new ArrayList();

private static void AddToArray(string firstName, string middleName, string lastName, int group, params Exam[] exams)

{

List<Exam> list = [.. exams];

array.Add(new Student(firstName, middleName, lastName, group, list));

}

private static void Main(string[] args)

{

AddToArray("Ivan", "Ivanovich", "Ivanov", 100, new Exam("Subject 1", 4), new Exam("Subject 2", 5), new Exam("Subject 3", 5));

AddToArray("Petr", "Petrovich", "Petrov", 100, new Exam("Subject 1", 4), new Exam("Subject 2", 5), new Exam("Subject 3", 5));

AddToArray("Student", "1", "1", 110, new Exam("Subject 1", 2), new Exam("Subject 2", 5), new Exam("Subject 3", 5));

AddToArray("Student", "2", "2", 112, new Exam("Subject 1", 4), new Exam("Subject 2", 5), new Exam("Subject 3", 5));

AddToArray("Student", "3", "3", 100, new Exam("Subject 1", 4), new Exam("Subject 2", 5));

AddToArray("Student", "4", "4", 110, new Exam("Subject 1", 3), new Exam("Subject 2", 3), new Exam("Subject 3", 3));

AddToArray("Student", "5", "5", 220, new Exam("Subject 1", 5), new Exam("Subject 2", 5), new Exam("Subject 3", 5));

ArrayList examed = new ArrayList();

foreach (Student student in array)

{

if (student.IsExamed())

{

examed.Add(student);

}

}

examed.Sort();

foreach (Student student in examed)

{

student.Print();

}

}

}

using System.Collections;

using System.Diagnostics.Contracts;

class Student : IComparable, IComparer {

readonly string firstName;

readonly string middleName;

readonly string lastName;

readonly int group;

List<Exam> exams;

public Student(string firstName, string middleName, string lastName, int group, List<Exam> exams)

{

this.firstName = firstName;

this.middleName = middleName;

this.lastName = lastName;

this.group = group;

this.exams = exams;

}

public int Compare(object? a, object? b)

{

if (a is not null && b is not null && a is Student st1 && b is Student st2)

{

return st1.group.CompareTo(st2.group);

}

throw new ArgumentException("ArgumentException");

}

public int CompareTo(object? b)

{

if (b is not null && b is Student another)

{

return this.group.CompareTo(another.group);

}

throw new ArgumentException("ArgumentException");

}

public void Print()

{

Console.Write(firstName + " " + middleName + " " + lastName + ", Group: " + group);

foreach (Exam exam in exams)

{

Console.Write(" " + exam.name + ": " + exam.grade);

}

Console.Write("\n");

}

public bool IsExamed()

{

if (exams.Count >= 3)

{

foreach (Exam exam in exams)

{

if (exam.grade <= 2) return false;

}

return true;

}

return false;

}

public static bool operator > (Student operand1, Student operand2)

{

return operand1.CompareTo(operand2) > 0;

}

public static bool operator < (Student operand1, Student operand2)

{

return operand1.CompareTo(operand2) < 0;

}

public static bool operator >= (Student operand1, Student operand2)

{

return operand1.CompareTo(operand2) >= 0;

}

public static bool operator <= (Student operand1, Student operand2)

{

return operand1.CompareTo(operand2) <= 0;

}

}

class Exam

{

public readonly string name;

public readonly int grade;

public Exam(string name, int grade)

{

this.name = name;

this.grade = grade;

}

}

Тестирование:

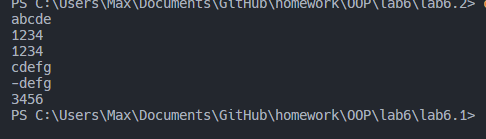


Рисунок 1 – Тест задачи 1

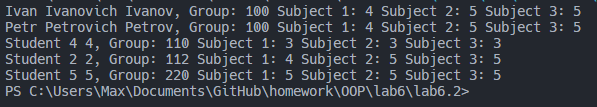


Рисунок 2 – Тест задачи 2

**Вывод:** узнал о реализации принципа полиморфизма с помощью интерфейсов, научился использовать пользовательские интерфейсы для задания поведения различных классов.