# **Практическая работа № 9. Интерфейсы**

Объём учебного времени – 2ч

Методические рекомендации

1. Цель работы: Научиться использовать пользовательские интерфейсы для задания поведения различных классов.
2. Перечень необходимых средств обучения:
   1. Технические средства обучения:

Компьютер Core i3 3.0, 2 Gb оперативной памяти, винчестер 250 Gb, DVD

* 1. Программное обеспечение:
* Программа ОС Windows XP/7;
* Антивирусные программы: Kaspersky AntiVirus;
* Браузер Internet Explorer;
* Интегрированная среда программирования MS Visual Studio 2017.

1. Основные теоретические положения:

Интерфейс является специальным видом классов. Синтаксис интерфейса аналогичен синтаксису класса:

interface <имя\_интерфейса> [:предки] <тело\_интерфейса> [;]

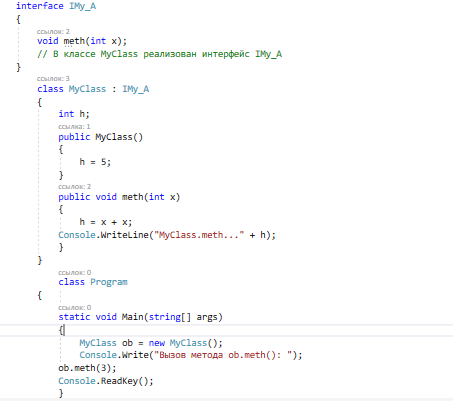
Интерфейс может наследовать свойства нескольких интерфейсов. Тело интерфейса составляют абстрактные методы, шаблоны свойств и индексаторов, а также события.

Интерфейс не может содержать константы, поля, операции, конструкторы, деструкторы, типы и любые статические элементы. В интерфейсе методы неявно являются открытыми (public-методами), при этом не разрешается явным образом указывать спецификатор доступа.

Чтобы реализовать интерфейс, нужно указать его имя после имени класса. В списке предков класса сначала указывается его базовый класс, если он есть, а затем через запятую – интерфейсы, которые реализует этот класс.

Методы, которые реализуют интерфейс, должны быть объявлены открытыми. Сигнатура типа в реализации метода должна в точности совпадать с сигнатурой типа, заданной в определении интерфейса. В классах, которые реализуют интерфейсы, можно определять дополнительные члены.

Пример 1:



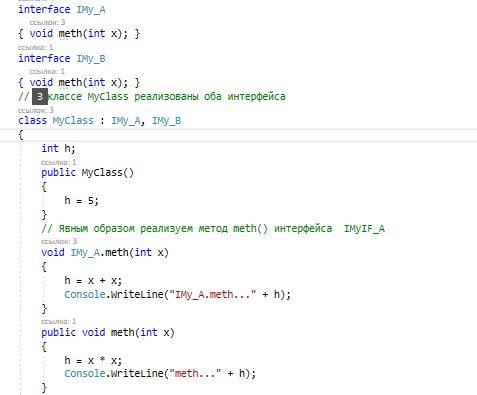
Можно объявить ссылочную переменную интерфейсного типа, которая может ссылаться на любой объект, реализующий ее интерфейс. При вызове метода для объекта посредством интерфейсной ссылки будет выполнена та версия указанного метода, которая реализована этим объектом.

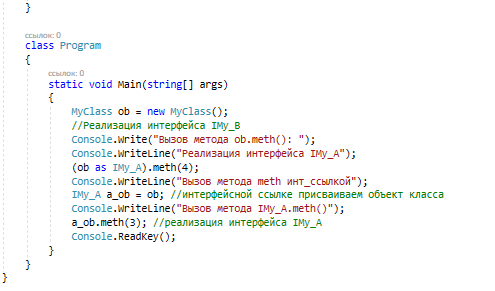
Элементы с одинаковыми именами или сигнатурой могут встречаться более чем в одном интерфейсе. В этом случае при множественном наследовании может возникнуть конфликт из-за неоднозначности ситуации, так как компилятор не может определить из контекста обращения к элементу, элемент какого именно из реализуемых интерфейсов требуется вызвать. Избежать неоднозначности можно с помощью бинарных операций is или as, которые позволяют убедиться, что объект поддерживает данный интерфейс или выполнить приведение к данному интерфейсу.

Для этой же цели можно использовать так называемую явную реализацию, когда имя интерфейса явно указывается перед реализуемым элементом через точку. Такая реализация интерфейса является закрытой. Спецификаторы доступа при этом не указываются. К таким элементам можно обращаться в программе только через объект типа интерфейса.

Кроме того, явная реализация позволяет избежать конфликтов при множественном наследовании, если элементы с одинаковыми именами или сигнатурой встречаются более чем в одном интерфейсе.

Пример 2. Реализовано два интерфейса, причем оба объявляют метод с одним именем.





Метод meth() имеет одинаковую сигнатуру в интерфейсах IMy\_A и IMy\_B. Поскольку единственный способ вызова явно задан- ного метода состоит в использовании интерфейсной ссылки, метод meth(), объявленный в интерфейсе IMy\_A, создает ссылку на интерфейс IMy\_A, а метод ob.meth(), вызывает метод, объявленный в интерфейсе IMy\_B.

**Наследование интерфейсов**

Интерфейс может не иметь или иметь сколько угодно интерфейсов-предков, в последнем случае он наследует все элементы всех своих базовых интерфейсов, начиная с самого верхнего уровня. Базовые интерфейсы должны быть доступны так же, как их потомки.

В интерфейсе-потомке можно также указать элементы, переопределяющие унаследованные элементы с такой же сигнатурой. В этом случае перед элементом указывается ключевое слово new.

Любой класс, который реализует интерфейс, должен реализовать все методы, определенные этим интерфейсом, включая методы, которые унаследованы от других интерфейсов.

Класс наследует все методы своего предка, в том числе те, которые реализовывали интерфейсы. Он может переопределить эти методы с помощью спецификатора new, но обращаться к ним можно будет только через объект класса. Если использовать для обращения ссылку на интерфейс, вызывается не переопределенная версия.

Однако если интерфейс реализуется с помощью виртуального метода класса, после его переопределения в потомке любой вариант обращения (через класс или через интерфейс) приведет к одному и тому же результату

Метод интерфейса, реализованный явным указанием имени, объявлять виртуальным запрещается.

Если класс наследует от класса и интерфейса, которые содержат методы с одинаковыми сигнатурами, унаследованный метод класса воспринимается как реализация интерфейса.

**Стандартные интерфейсы среды .NET Framework**

В библиотеке классов .NET определено множество стандартных интерфейсов, задающих желаемое поведение объектов. Стандартные интерфейсы поддерживаются многими стандартными классами биб- лиотеки. Можно создавать и собственные классы, поддерживающие стандартные интерфейсы, что позволит использовать объекты этих классов стандартными способами.

Во многих классах необходимо реализовать интерфейс IComparable или IComparer, поскольку они позволяют сравнить два объекта.

Интерфейс IComparable состоит только из одного метода:

int CompareTo(object v)

Этот метод сравнивает вызывающий объект со значением параметра v. Метод возвращает положительное число, если вызывающий объект больше объекта v, нуль, если два сравниваемых объекта равны, и отрицательное число, если вызывающий объект меньше объекта

Метод CompareTo может сгенерировать исключение типа ArgumentException, если тип объекта v несовместим с вызывающим объектом.

В интерфейсе IComparer определен метод Compare (), который позволяет сравнивать два объекта:

int Compare(object vl, object v2)

Метод Compare() возвращает положительное число, если значение vl больше значения v2, отрицательное, если vl меньше v2, и нуль, если сравниваемые значения равны. Этот интерфейс можно использовать для задания способа сортировки элементов коллекции.

1. Практические указания:
   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:

* реализовать в классе интерфейсы при условии наследования базового класса (абстрактного);
* выполнить явную реализацию членов интерфейса;
* вызов методов для объекта посредством интерфейсной переменной.
  1. Модифицировать проект, используя явную реализацию интерфейса для методов с одинаковой сигнатурой в классе 1.

**Варианты заданий:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **интерфейсы** | **поле** | **классы** | **F0 F1 возвращают:** | |
| неявная реализация | неявная реализация |
| 1. | interface Iy  {void F0(double параметр);  void F1();}  interface Iz  {void F0(double параметр);  float F1(int параметр);} | double | 1 |  | w2+5 |
| 2 | ew | - |
| 2. | interface IA  {void F0(out double параметр);  int F1(double параметр);} interface IB  {void F0( out double параметр);  void F1();} | double | 1 | w2 | 15/w |
| 2 | w3 | - |
| 3. | interface Ix  {char F0();  void F1(ref char параметр);}  interface Iy  {void F0(char параметр);  void F1(ref char  параметр);} | char | 1 | Символ, преобразо- ванный в нижний  регистр | цифру ‘5’, если символ буква |
| 2 | \*, если символ – буква | - |
| 4. | interface Ix  {void F0(); void F1(string параметр);} interface Iy  {void F0(string параметр); void F1(string параметр);} | string | 1 | Строку, удалив два первых символа | Строку, заменив первый символ символом  ‘-‘ |
| 2 | Строку, удалив два последних  символа | - |
| 5 | interface Ix  {bool F0(char параметр);  void F1(int параметр);} interface Iy  {bool F0(char параметр); void F1(char параметр);} | char | 1 | Определяет, является ли символ (код символа) цифрой | Определяет, содержится ли значение кода символа в диапазоне кодов ASCII |
| 2 | Определяет, является ли символ знаком  препинания | - |
| 6. | interface Ix  {string F0(int параметр);  void F1(int параметр);} interface Iy  {void F0(int параметр,out string параметр);  void F1(int параметр);} | string | 1 | Удвоенную строку с удаленным  k-м символом | Cтроку, заменив k- ый символ знаком + |
| 2 | подстроку, начиная с k-ей позиции | - |
| 7. | interface Iy  {void F0(double параметр);  void F1(double параметр);}  interface Iz  {void F0(double параметр);  double F1();} | double | 1 | sin(w) | w+2 |
| 2 | 2/w\*ln(w) | - |
| 8. | interface Ix  {void F0(int k);  void F1(int k, out int параметр);}  interface Iy  {void F0(int k);  void F1(int k);} | int | 1 | k\*w2 | 2w-k |
| 2 | |w|-k | - |
| 9. | interface IA  {double F0(double параметр);  int F1(double параметр);} interface IB  {double F0(double параметр);  void F1();} | double | 1 | 1/ew | w/10 |
| 2 | w-1010\*w | - |
| 10. | interface Ix  {int F0();  void F1();}  interface Iy  {int F0();  void F1(char параметр);} | char | 1 | Если символ является цифрой найти его числовое значение, иначе его код | вернуть код символа |
| 2 | Если символ является буквой х, вывести  число 0 | - |
| 11. | interface Iy  {void F0(double параметр);  float F1(int параметр);} interface Iz  {void F0(out double параметр);  Float F1(int параметр);} | float | 1 | 7w/ln(2) | 7w\*2.6 |
| 2 | |w-10| | - |
| 12. | interface Is  {string F0();  void F1();}  interface Ir  {void F0(string параметр); void F1();} | string | 1 | Заменить все буквы в строке на прописные символы | изменить регистр всех букв в строке |
| 2 | Заменить на пробелы знаки пре-  пинания | - |

* 1. 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.

Варианты заданий:

* 1. Составить багажную ведомость камеры хранения, включив следующие данные: ФИО пассажира, количество вещей, общий вес вещей. Вывести в новый список информацию о тех пассажирах, средний вес багажа которых превышает заданный, отсортировав их по количеству вещей, сданных в камеру хранения.
  2. Составить список студентов, включающий ФИО, курс, группу, дату и 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.
  3. Составить инвентарную ведомость склада, включив следующие данные: вид продукции, стоимость, сорт, количество. Вывести в новый список информацию о той продукции, количество которой меньше заданной величины, отсортировав ее по количеству продукции на складе.
  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.
  5. Составить список вкладчиков банка, включив следующие данные: ФИО, № счета, сумма, дату открытия счета. Вывести в новый список информацию о тех вкладчиках, которые открыли вклад в текущем году, отсортировав их по сумме вклада.
  6. Составить автомобильную ведомость, включив следующие данные: марка автомобиля, фамилия его владельца, год приобретения, пробег. Вывести в новый список информацию об автомобилях, выпущенных ранее определенного года, отсортировав их по пробегу.
  7. Составить инвентарную ведомость игрушек, включив следующие данные: название игрушки, ее стоимость (в руб.), возрастные границы детей, для которых предназначена игрушка. Вывести в новый список информацию о тех игрушках, которые предназначены для детей от N до M лет, отсортировав их по стоимости.
  8. Составить список студентов группы, включив следующие данные: ФИО, дату рождения, какую школу окончил. Вывести в новый список информацию о студентах, окончивших заданную школу, отсортировав их по году рождения.
  9. В библиотеке имеется список книг. Каждая запись этого списка содержит фамилии авторов, название книги, год издания, цена экземпляра. Определить, имеются ли в данном списке книги, в названии которых встречается некоторое ключевое слово (например, "программирование"). Если имеются, то вывести фамилии авторов, название и год издания всех таких книг.
  10. Составить список студентов, включающий фамилию, факультет, курс, группу, 5 оценок. Вывести в новый список информацию о тех студентах, которые имеют хотя бы одну двойку, отсортировав их по курсу.
  11. Составить список больных отделения, включив следующие данные: ФИО, болезнь, дата поступления, рабочий стаж. Вывести в новый список информацию о больных, находящихся на лечении больше недели, отсортировав их по ФИО.
  12. В пресс-центре выставки программных средств хранятся данные о каждом экспонате: название, автор, количество заявок на него. Вывести в новый список экспонаты, получившие больше трех заявок, отсортировав по числу заявок.
  13. Дополнительные задания

1. Напишите программу, содержащую абстрактный базовый класс с защищенным полем, являющимся ссылкой на целочисленный массив. У класса должен быть конструктор с одним аргументом (определяет размер массива и создает его), целочисленное свойство (значение - размер массива), абстрактный метод (без аргументов, не возвращает результат) и индексатор с целочисленным индексом (доступен для чтения и записи). В производном классе описать абстрактный метод из базового класса, чтобы он отображал в консоли содержимое массива. Индексатор определить так, чтобы с его помощью можно было прочитать значение элемента массива и присвоить значение элементу массива.
2. Напишите программу, содержащую абстрактный класс с двумя защищенными целочисленными полями и конструктор с двумя целочисленными аргументами. В классе должен быть объявлен абстрактный индексатор с целочисленным индексом. Опишите интерфейс, в котором есть метод с целочисленным аргументом и целочисленным результатом. Опишите класс, который наследует абстрактный базовый класс и реализует интерфейс. В этом классе опишите индексатор так, чтобы при четном индексе выполнялось обращение к первому полю, а при нечетном индексе обращение выполнялось ко второму полю. Метод следует описать таким образом, чтобы он результатом возвращал сумму значений полей, умноженную на аргумент метода.
3. Напишите программу, содержащую абстрактный класс и два интерфейса. Класс должен содержать объявление абстрактного свойства (с двумя аксессорами), абстрактного индексатора (с двумя аксессорами) и абстрактного метода. Такое же свойство, индексатор и метод должны быть в интерфейсах. На основе абстрактного класса и интерфейсов необходимо создать класс. В этом классе необходимо выполнить явную реализацию для свойства, индексатора и метода для каждого из интерфейсов. Проверьте работу свойства, индексатора и метода, получив доступ к объекту класса через объектную переменную и через интерфейсные переменные.
4. Контрольные вопросы
5. Свойство это…
6. Как описывается интерфейс? Его назначение.
7. Какие члены может содержать интерфейс?
8. Какие спецификаторы допустимы у методов, реализующих интерфейс?
9. В каких случаях используется явная реализация интерфейса?
10. Как осуществляется наследование интерфейсов?
11. Можно ли явно реализованные методы объявлять виртуальными?
12. Можно ли повторно реализовать интерфейс, указав его имя в списке предков класса наряду с классом-предком?
13. Какие стандартные интерфейсы используются для работы с коллекциями?
14. Список рекомендуемой литературы:

**Основная литература:**

1. Немцова Т.И. Програмиирование на языке выского уровня. Программирование на языке Object Pascal: учеб. пособие /Т.И. Немцова, С.Ю. Голова, И.В. Абрамова. – М.: ФОРУМ: ИНФРА – М, 2012. – 496 с.
2. Овечкин Г.В. Компьютерное моделирование: учебник для студ. учреждений сред. проф. образования / Г.В. Овечкин.- М.: Издательский центр «Академия», 2015.-224с.
3. Немцова, Т. И.  Программирование на языке высокого уровня. Программирование на языке С++ : учеб. пособие / Т. И. Немцова, С. Ю. Голова, А. И. Терентьев. - М. : ФОРУМ : ИНФРА-М, 2012. - 512 с. : ил. + CD.
4. Гуриков, С. Р. Введение в программирование на языке Visual C# : учеб. пособие / С. Р. Гуриков. - М. : ФОРУМ : ИНФРА-М, 2013. - 448 с.

**Дополнительная литература:**

1. Рао Сиддхартха Освой самостоятельно С++ за 21 день, 7 изд.: Пер с англ.-М.: ООО «И.Д.Вильямс», 2013 – 688 с.: ил. – Парал.тит.англ.
2. Голицына О.Л. Программное обеспечение: учеб. пособие для среднего профессионального образования-М.:ФОРУМ;ИНФРА-М,2006.-432 с.
3. Виллемер А. Программирование на С++/А. Виллемер;[пер. с нем. М.А.Райтман].-М.:Эксмо,2013.-528с.+CD.-(Мировой компьютерный бестселлер).
4. Культин Н.Б. Microsoft Visual C++ в задачах и примерах.- СПб.: БХВ-Петербург,2010.-272 с.:ил.+CD-ROM.
5. Партыка Т.Л. Операционные системы, среды и оболочки: учеб. пособие для студ. учреждений сред. проф. Образования/ Т.Л. Партыка, И.И. Попов. – 3-е изд., перераб. и доп. – М.: ФОРУМ, 2010. – 543 с.
6. Окулов С.М. Основы программирования: учебное пособие.- М.:БИНОМ. Лаборатория знаний, 2010.- 440 с.
7. Голицына О.Л. Основы алгоритмизации и программирования: учеб. пособие для сред. проф. образования.- М: ФОРУМ; ИНФРА-М,2005.-432 с.
8. Рихтер Дж. Программирование приложений для Microsoft Windows /Пер. с англ. – M.: Microsoft Press, 2003. – C.48-313.