

ЛАБОРАТОРНАЯ РАБОТА 12. ПОСТРОЕНИЕ ИЕРАРХИИ КЛАССОВ

1. Цель и содержание

Цель лабораторной работы: изучить механизм организации наследования классов.

Задачи лабораторной работы:

- научиться объявлять производные классы;
- научиться создавать иерархии классов;
- научиться использовать механизм полиморфизма.

2. Формируемые компетенции

Лабораторная работа направлена на формирование следующих компетенций:

- способность к проектированию базовых и прикладных информационных технологий (ПК-11);
- способность разрабатывать средства реализации информационных технологий (методические, информационные, математические, алгоритмические, технические и программные) (ПК-12).

3. Теоретическая часть

2.1 Наследование реализации. Наследование реализации (implementation inheritance) означает, что тип происходит от базового типа, получая от него все поля-члены и функции-члены.

Синтаксис наследования реализации:

```
class ПроизводныйКласс : БазовыйКласс
{
    // Данные–члены и функции–члены
}
```

Если при определении класса не указан базовый класс, то С# предполагает, что базовым классом является System.Object.

2.2 Создание иерархии классов. При наследовании реализации производный класс наследует реализацию каждой функции базового типа, если только в его определении не указано, что реализация функции должна быть переопределена.

Определим следующую иерархию классов (рис. 17.1) и продемонстрируем, как наследуется реализация и как переопределяются свойств и методов.



Рисунок 17.1 – Иерархия классов.

Создадим код, описывающий данную иерархию. Определим базовый класс:

```

class Человек
{
    public Человек(string Ф, string И, string О, int Возр)
    {
        Фамилия = Ф; Имя = И; Отчество = О; Возраст = Возр;
    }

    public Человек()
    {
        Фамилия = "нет данных"; Имя = ""; Отчество = "";
        Возраст = 18;
    }

    public string Фамилия, Имя, Отчество;
    private DateTime ДатаРождения;

    public virtual string ФИО
    {
        get { return Фамилия + " " + Имя + " " + Отчество; }
    }

    public int Возраст
    {
        get { return DateTime.Now.Year - ДатаРождения.Year; }
        set
        {
            int ГодРождения = DateTime.Now.Year - value;

            ДатаРождения = Convert.ToDateTime(ГодРождения.ToString()+".01.01");
        }
    }
}

```

Обратите внимание на наличие двух конструкторов, механизм хранения возраста человека и ключевое слово `virtual` у свойства «ФИО». Ключевое слово `virtual` указывает, что данное свойство будет переопределено в производном классе.

Определим производный класс «Учитель»:

```

class Учитель: Человек
{
    public Учитель()
    :base()
    {
        УченоеЗвание = УченыеЗвания.Без_Звания;
        УченаяСтепень = УченыеСтепени.Без_Степени;
    }

    public Учитель(string Ф, string И, string О, int Возр, УченыеЗвания УЗ, УченыеСтепени УС)
    :base(Ф, И, О, Возр)
    {
        УченоеЗвание = УЗ;
        УченаяСтепень = УС;
    }

    public УченыеЗвания УченоеЗвание;
    public УченыеСтепени УченаяСтепень;

    public override string ФИО
    {
        get
        {
            return УченаяСтепень.ToString() + ", " + УченоеЗвание.ToString() + ", " + base.ФИО;
        }
    }
}

```

Следует обратить внимание на использование ключевого слова `override` у свойства «ФИО», которое указывает на то, что данное свойство имеет новую реализацию, отличающуюся от реализации базового класса.

Определим второй производный класс «Студент»:

```

class Студент: Человек
{
    public Студент(string Ф, string И, string О, int Возр, Специальности Спец)
    :base(Ф, И, О, Возр)
    {
        Специальность = Спец;
    }

    public Специальности Специальность;

    public override string ФИО
    {
        get
        {
            return base.ФИО + ", " + Специальность.ToString();
        }
    }
}

```

В обоих производных классах следует обратить внимание на реализацию конструкторов производных классов, использование ключевого слова `base` в коде свойства «ФИО» и при объявлении конструкторов.

Также интерес представляют типы данных, используемые для членов-данных: «Специальности», «УченыеЗвания», «УченыеСтепени». Вот определения данных типов-перечислений:

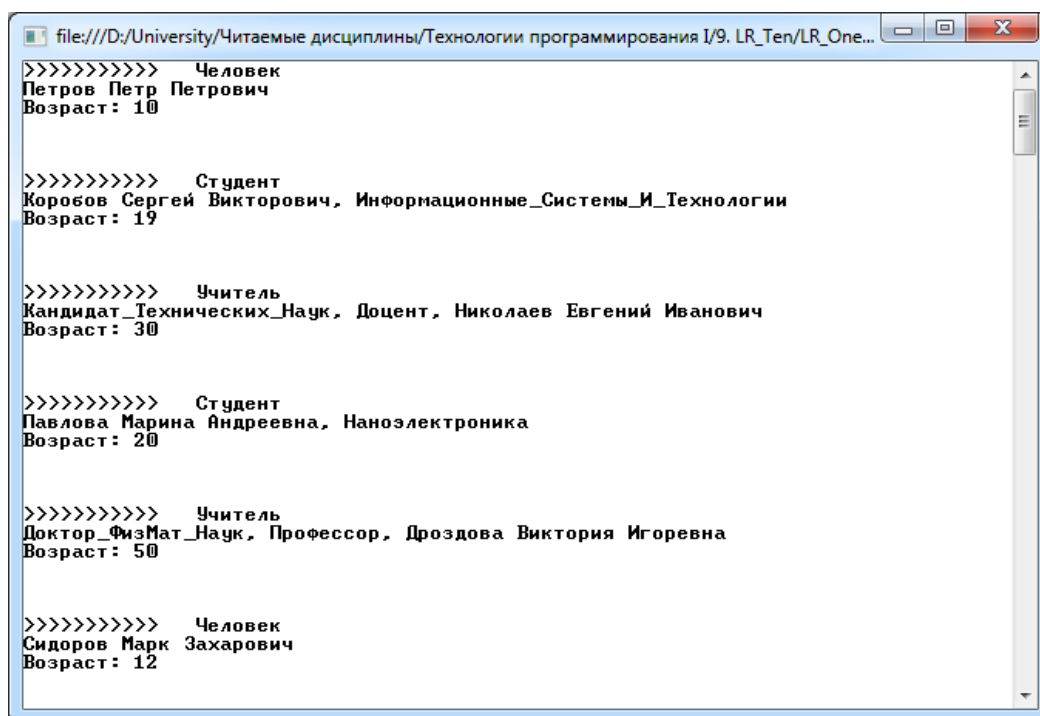
```
public enum УченыеЗвания
{
    Доцент,
    Профессор,
    Академик,
    Без_Звания
}

public enum УченыеСтепени
{
    Кандидат_Технических_Наук,
    Кандидат_ФизМат_Наук,
    Кандидат_Педагогических_Наук,
    Доктор_ФизМат_Наук,
    Без_Степени
}

public enum Специальности
{
    Информационные_Системы_И_Технологии,
    Безопасность_Информационных_Систем,
    Технология_Защиты_Информации,
    Психология,
    Нанoeлектроника
}
```

Наконец, продемонстрируем использование объявленной иерархии классов.

В программе объявим массив объектов типа «Человек». Следует понимать, что при объявлении такого массива, его элементам можно присваивать объекты любого производного класса, причем каждый объект будет вести себя по-своему (за счет переопределения свойств и методов).



Полная диаграмма типов в полученном приложении показана на рис. 17.2.

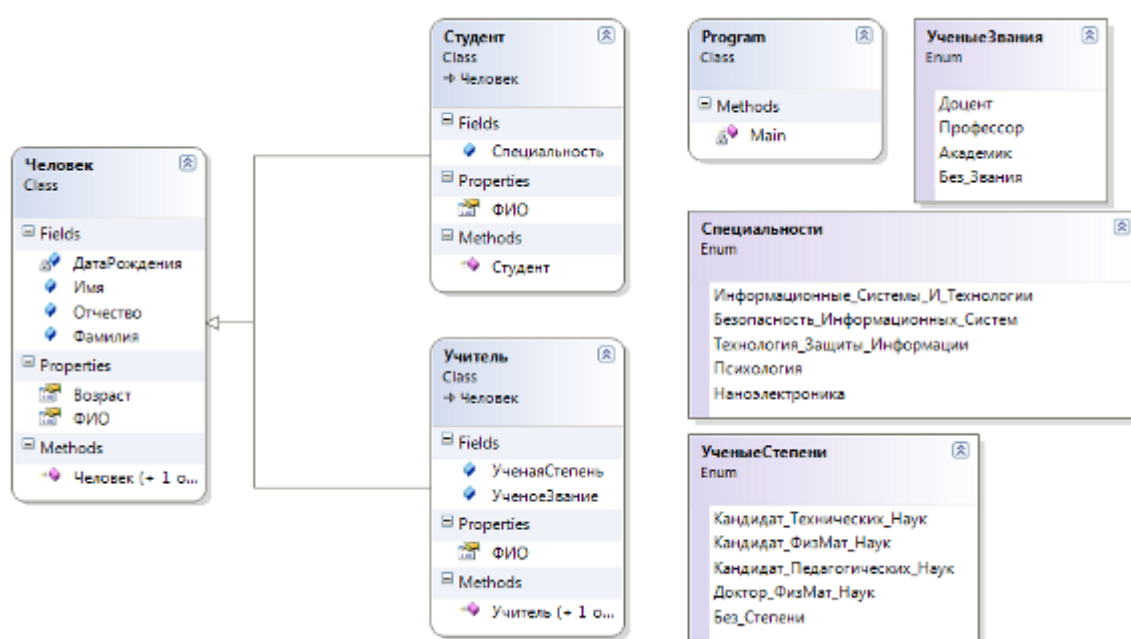


Рисунок 17.2 – Диаграмма типов приложения (создана средствами VS).

Структуры всегда наследуются от `System.ValueType`. Они могут также наследовать любое количество интерфейсов. Классы всегда наследуются от

одного класса по вашему выбору. Они также могут наследовать любое количество интерфейсов.

4. Оборудование и материалы

Для выполнения лабораторной работы рекомендуется использовать персональный компьютер со следующими характеристиками: 64-разрядный (x64) процессор с тактовой частотой 1 ГГц и выше, оперативная память – 1 Гб и выше, свободное дисковое пространство – не менее 1 Гб, графическое устройство DirectX 9. Программное обеспечение: операционная система WINDOWS 7 и выше, Microsoft Visual Studio 20112 и выше.

5. Указания по технике безопасности

Техника безопасности при выполнении лабораторной работы определяется общепринятой для пользователей персональных компьютеров. Самостоятельно не производить ремонт персонального компьютера, установку и удаление программного обеспечения; в случае неисправности персонального компьютера сообщить об этом обслуживающему персоналу лаборатории; не касаться электрических розеток металлическими предметами; рабочее место пользователя персонального компьютера должно содержаться в чистоте; не разрешается возле персонального компьютера принимать пищу, напитки.

6. Методика и порядок выполнения работы

1. Создайте консольное приложение в соответствии с алгоритмом, представленным в лабораторной работе №1.

2. Изучите пример создания иерархии классов, представленный в разделе «Теоретическое обоснование» данной лабораторной работы.

3. Постройте свою иерархию классов в соответствии с индивидуальным заданием. В результате выполнения лабораторной работы должны быть реализованы следующие механизмы:

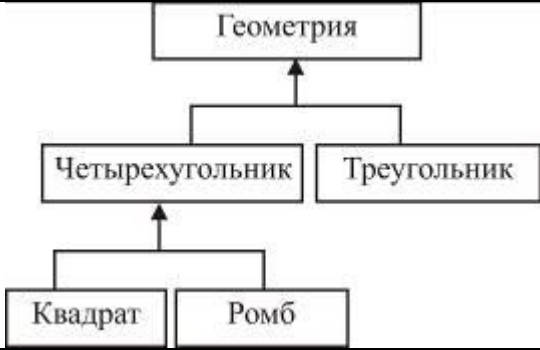





- использование типа-перечисления (хотя бы одного);
- использование переопределенного свойства (хотя бы одного);
- использование переопределенного метода (хотя бы одного);
- использование вызова базового конструктора;
- использование вызова любого базового метода (отличного от конструктора).

4. Продемонстрируйте использование классов, созданной иерархии (легче всего это сделать с использованием массивов). При защите работы укажите признаки присутствия полиморфного поведения в программе (реализация полиморфизма).


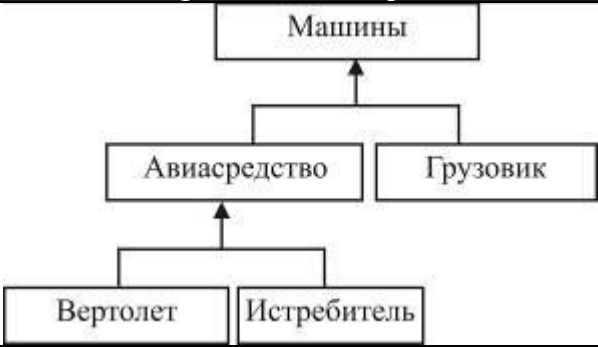





Индивидуальное задание.

Спроектируйте класс, наполните его требуемой функциональностью, продемонстрируйте работоспособность класса.

Вариант	Иерархия классов
1	<pre> graph BT ГрузовойАвто --> Автомобиль ЛегковойАвто --> Автомобиль Автомобиль --> ТехСредство Вертолёт --> ТехСредство </pre>
2	<pre> graph BT Сервер --> КомпТехника Ноутбук --> КомпТехника ПК --> КомпТехника </pre>

Вариант	Иерархия классов
3	 <pre> classDiagram class Геометрия class Четырехугольник class Треугольник class Квадрат class Ромб Геометрия < -- Четырехугольник Геометрия < -- Треугольник Четырехугольник < -- Квадрат Четырехугольник < -- Ромб </pre>
4	 <pre> classDiagram class Издание class Книга class Журнал class ЭлРесурс Издание < -- Книга Издание < -- Журнал Издание < -- ЭлРесурс </pre>
5	 <pre> classDiagram class Мультимедиа class Изображение class Видео class Растровое class Векторное Мультимедиа < -- Изображение Мультимедиа < -- Видео Изображение < -- Растровое Изображение < -- Векторное </pre>
6	 <pre> classDiagram class Вооружение class Стрелковое class Танк class Сомолет Вооружение < -- Стрелковое Вооружение < -- Танк Вооружение < -- Сомолет </pre>
7	 <pre> classDiagram class Машины class Авиасредство class Грузовик class Вертолет class Истребитель Машины < -- Авиасредство Машины < -- Грузовик Авиасредство < -- Вертолет Авиасредство < -- Истребитель </pre>
8	 <pre> classDiagram class Растение class Трава class Дерево class Кустарник Растение < -- Трава Растение < -- Дерево Растение < -- Кустарник </pre>

Вариант	Иерархия классов
9	<pre> graph BT ПО[ПО] ОС[ОС] Прикладное[Прикладное] СвободныеОС[СвободныеОС] ПлатныеОС[ПлатныеОС] ПО --> ОС ПО --> Прикладное ОС --> СвободныеОС ОС --> ПлатныеОС </pre>
10	<pre> graph BT ЭВМ[ЭВМ] Мобильный[Мобильный] Стационарный[Стационарный] Планшет[Планшет] Ноутбук[Ноутбук] ЭВМ --> Мобильный ЭВМ --> Стационарный Мобильный --> Планшет Мобильный --> Ноутбук </pre>
11	<pre> graph BT Электроника[Электроника] СотТелефон[СотТелефон] Фотоаппарат[Фотоаппарат] Планшет[Планшет] Электроника --> СотТелефон Электроника --> Фотоаппарат Электроника --> Планшет </pre>
12	<pre> graph BT ТехСредство[ТехСредство] Автомобиль[Автомобиль] Вертолёт[Вертолёт] ГрузовойАвто[ГрузовойАвто] ЛегковойАвто[ЛегковойАвто] ТехСредство --> Автомобиль ТехСредство --> Вертолёт Автомобиль --> ГрузовойАвто Автомобиль --> ЛегковойАвто </pre>
13	<pre> graph BT БытТехника[БытТехника] Телевизор[Телевизор] Утюг[Утюг] Микроволновка[Микроволновка] БытТехника --> Телевизор БытТехника --> Утюг БытТехника --> Микроволновка </pre>
14	<pre> graph BT ЭВМ[ЭВМ] Мобильный[Мобильный] Стационарный[Стационарный] Планшет[Планшет] Ноутбук[Ноутбук] ЭВМ --> Мобильный ЭВМ --> Стационарный Мобильный --> Планшет Мобильный --> Ноутбук </pre>

Вариант	Иерархия классов
15	 <pre> graph BT Ручка --> КанцТовары Органайзер --> КанцТовары Скоросшиватель --> КанцТовары </pre>
16	 <pre> graph BT Вертолет --> Авиасредство Истребитель --> Авиасредство Авиасредство --> Машины Грузовик --> Машины </pre>
17	 <pre> graph BT Квартира --> Недвижимость Дом --> Недвижимость Гараж --> Недвижимость </pre>
18	 <pre> graph BT Трактор --> Тяжелое Комбайн --> Тяжелое Тяжелое --> Машиностроение Среднее --> Машиностроение </pre>
19	 <pre> graph BT Фотон --> ЭлемЧастица Электрон --> ЭлемЧастица Протон --> ЭлемЧастица </pre>
20	 <pre> graph BT Прямоугольник --> Многоугольник Треугольник --> Многоугольник Многоугольник --> Геометрия Эллипс --> Геометрия </pre>
21	 <pre> graph BT Программист --> Специалист Врач --> Специалист Летчик --> Специалист </pre>

Вариант	Иерархия классов
22	<pre> graph BT ПО[ПО] --> ОС[ОС] ПО --> Прикладное[Прикладное] ОС --> СвободныеОС[СвободныеОС] ОС --> ПлатныеОС[ПлатныеОС] </pre>
23	<pre> graph BT БанкКарта[БанкКарта] --> Накопительная[Накопительная] БанкКарта --> Платежная[Платежная] БанкКарта --> Зарплатная[Зарплатная] </pre>
24	<pre> graph BT Сервис[Сервис] --> Продажи[Продажи] Сервис --> Лизинг[Лизинг] Сервис --> Кредит[Кредит] </pre>
25	<pre> graph BT Мебель[Мебель] --> Стол[Стол] Мебель --> МягкМебель[МягкМебель] Мебель --> Шкаф[Шкаф] </pre>

7. Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы.
2. Цели лабораторной работы.
3. Ответы на контрольные вопросы.
4. Экранные формы и листинг программного кода, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения.

Отчет о выполнении лабораторной работы в письменном виде сдается преподавателю.

8. Контрольные вопросы

1. Что такое наследование реализации? Как описать синтаксически наследование реализации?
2. Для чего используется ключевое слово `base`?
3. Можно ли переопределить метод класса? Свойства класса? Данные класса?
4. Как переопределить метод в производном классе?
5. Для чего используется ключевое слово `virtual`?
6. Для чего используется ключевое слово `override`?
7. Как поменять цвет фона в консольном приложении?
8. Как построить диаграмму типов данных, используемых в приложении средствами Visual Studio 2008/ 2010?
9. Сколько базовых классов может быть у любого класса в C#?

9. Список литературы

Для выполнения лабораторной работы, при подготовке к защите, а также для ответа на контрольные вопросы рекомендуется использовать следующие источники: [5-7].

ЛАБОРАТОРНАЯ РАБОТА 13. РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ

1. Цель и содержание

Цель лабораторной работы: изучить принципы работы с типами интерфейсов в C#.

Задачи лабораторной работы:

- научиться объявлять интерфейсы в C#;
- научиться создавать классы, реализующие интерфейсы;

2. Формируемые компетенции

Лабораторная работа направлена на формирование следующих компетенций:

- способность к проектированию базовых и прикладных информационных технологий (ПК-11);
- способность разрабатывать средства реализации информационных технологий (методические, информационные, математические, алгоритмические, технические и программные) (ПК-12).

3. Теоретическая часть

2.1 Наследование интерфейсов. Кроме наследования реализации (implementation inheritance) в языке C# реализован механизм наследования интерфейсов (interface inheritance). Необходимо понимать, что не все объектно-ориентированные языки поддерживают интерфейсы.

Если класс наследует интерфейс, класс как бы берет на себя обязательства реализовать некоторый функционал.