Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра инфокоммуникаций

**«ПРОЕКТИРОВАНИЕ ИЕРАРХИИ КЛАССОВ»**

**ОТЧЕТ**

**по лабораторной работе №8**

**дисциплины**

**«Технологии программирования»**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнил:  Мизин Глеб Егорович  2 курс, группа ПИЖ-б-о-21-1,  09.03.04 «Программная инженерия», направленность (профиль) «Разработка и сопровождение программного обеспечения», очная форма обучения  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | | Проверил:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | |  | |

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_ Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ставрополь, 2022 г.

**Проработка учебной задачи**:

namespace Helloworld

{

    class Program

    {

        static void Main(string[] *args*)

        {

*//Объявим массив людей*

            Человек[] mas = new Человек[6];

*//Заполним массив значениями разных (!) типов*

            mas[0] = new Человек("Петров", "Петр", "Петрович", 10);

            mas[1] = new Студент("Коробов", "Сергей", "Викторович", 19, Специальности.Информационные\_Системы\_И\_Технологии);

            mas[2] = new Учитель("Николаев", "Евгений", "Иванович", 30, УчённыеЗвания.Доцент, УчённыеСтепени.Кандитат\_Технических\_Наук);

            mas[3] = new Студент("Павлова", "Марина", "Андреевна", 20, Специальности.Наноэлектроника);

            mas[4] = new Учитель("Дроздова", "Виктория", "Игоревна", 50, УчённыеЗвания.Профессор, УчённыеСтепени.Доктор\_Физмат\_Наук);

            mas[5] = new Человек("Сидоров", "Марк", "Захарович", 12);

*//Вывод информации*

            for (int i = 0 ; i < mas.Length; i++)

            {

                Console.WriteLine(">>>>>>>>>>>>>     " + mas[i].GetType().Name);

                Console.WriteLine(mas[i].ФИО);

                Console.WriteLine("Возраст: " + mas[i].Возраст.ToString()+"\n");

                Console.WriteLine("\n");

            }

            Console.ReadKey();

        }

    }

    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");

            }

        }

    }

    class Учитель: Человек

    {

        public Учитель()

            :base()

        {

            УчённоеЗвание = УчённыеЗвания.Без\_Звания;

            УчённаяСтепень = УчённыеСтепени.Без\_Степени;

        }

        public Учитель (String *Ф*, String *И*, String *О*, int *Возр*, УчённыеЗвания *УЗ*, УчённыеСтепени *УС*)

            :base(*Ф*, *И*, *О*, *Возр*)

        {

            УчённоеЗвание = *УЗ*;

            УчённаяСтепень = *УС*;

        }

            public УчённыеЗвания УчённоеЗвание;

            public УчённыеСтепени УчённаяСтепень;

        public override string ФИО

        {

            get

            {

                return УчённаяСтепень.ToString() + ", " + УчённоеЗвание.ToString() + ", " + base.ФИО;

            }

        }

    }

    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();

            }

        }

    }

    public enum УчённыеЗвания

    {

        Доцент,

        Профессор,

        Академик,

        Без\_Звания

    }

    public enum УчённыеСтепени

    {

        Кандитат\_Технических\_Наук,

        Кандидат\_ФизМат\_Наук,

        Кандидат\_Педагогических\_Наук,

        Доктор\_Физмат\_Наук,

        Без\_Степени

    }

    public enum Специальности

    {

        Информационные\_Системы\_И\_Технологии,

        Безопасность\_Информацуонных\_Систем,

        Технология\_Защиты\_Информации,

        Психология,

        Наноэлектроника

    }

}

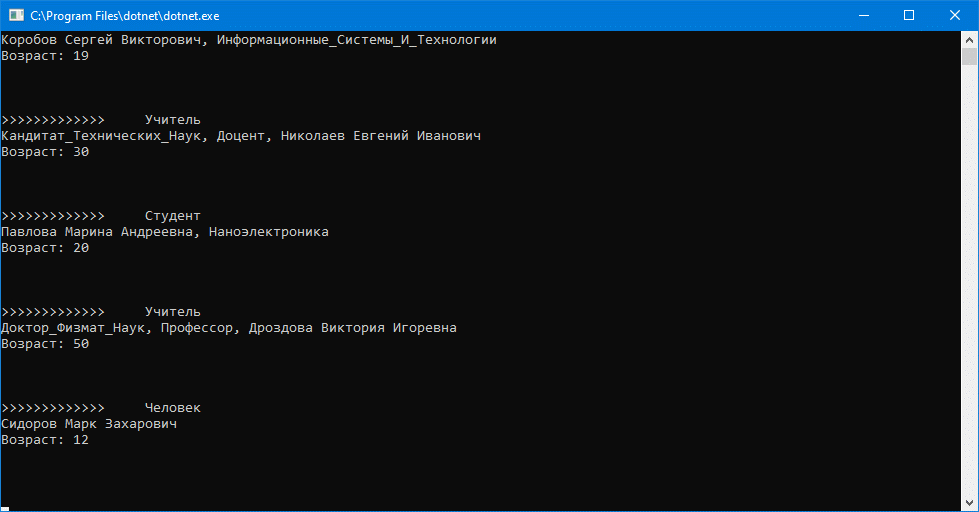


Рисунок 1 – Результат работы программы из примера

**Индивидуальное задание**:

namespace Helloworld

{

    class Program

    {

        static void Main(string[] *args*)

        {

            ТехСредство[] mas = new ТехСредство[4];

            mas[0] = new ТехСредство(СтранаПроизводитель.Россия, МаркаТехСред.Лада, 2009);

            mas[1] = new Автомобиль(СтранаПроизводитель.Германия, МаркаТехСред.Опель, 2008, "Автомат");

            mas[2] = new Легковое(СтранаПроизводитель.Италия, МаркаТехСред.Ферари, 1990, "Автомат", 240, Тип.Универсал);

            mas[3] = new Грузовое(СтранаПроизводитель.Америка, МаркаТехСред.Нисан, 2010, "Механика" , 90, 2000);

            for (int i = 0 ; i < mas.Length; i++)

            {

                Console.WriteLine(">>>          " + mas[i].GetType().Name);

                Console.WriteLine(mas[i].База);

                Console.WriteLine("\n");

            }

            Console.ReadKey();

        }

    }

    class ТехСредство

        {

            public ТехСредство (СтранаПроизводитель *С*, МаркаТехСред *М*, int *Год*)

            {

                Страна = *С*; Марка = *М*; ГодПроизводства = *Год*;

            }

            public ТехСредство()

            {

                Страна = СтранаПроизводитель.Нет\_данных; Марка = МаркаТехСред.Нет\_данных; ГодПроизводства = 0;

            }

            public СтранаПроизводитель Страна;

            public МаркаТехСред Марка;

            private int ГодПроизводства;

            public virtual string База

            {

                get {return Страна + " " + Марка + " " + ГодПроизводства;}

            }

        }

        class Автомобиль: ТехСредство

        {

            public Автомобиль (СтранаПроизводитель *С*, МаркаТехСред *М*, int *Год*, string *КП*)

                :base(*С*, *М*, *Год*)

            {

                КоробкаПередачь = *КП*;

            }

            public string КоробкаПередачь;

            public override string База

            {

                get

                {

                    return КоробкаПередачь.ToString() + ", " + base.База;

                }

            }

        }

        class Вертолёт: ТехСредство

        {

            public Вертолёт (СтранаПроизводитель *С*, МаркаТехСред *М*, int *Год*, int *Лоп*)

                :base(*С*, *М*, *Год*)

            {

                Лопости = *Лоп*;

            }

            public int Лопости;

            public override string База

            {

                get

                {

                    return "Размер лопости: " + Лопости + ", " + base.База;

                }

            }

        }

        class Легковое: Автомобиль

        {

            public Легковое (СтранаПроизводитель *С*, МаркаТехСред *М*, int *Год*, string *КП*, int *ЛС*, Тип *Т*)

                :base(*С*, *М*, *Год*, *КП*)

            {

                МД = *ЛС*;

                type = *Т*;

            }

            public int МД;

            public Тип type;

            public override string База

            {

                get

                {

                    return "Мощьность двигателя: " + МД + " Л.С. " + type + base.База;

                }

            }

        }

        class Грузовое: Автомобиль

        {

            public Грузовое (СтранаПроизводитель *С*, МаркаТехСред *М*, int *Год*, string *КП*, int *ЛС*, int *ПеревозВес*)

                :base(*С*, *М*, *Год*, *КП*)

            {

                МД = *ЛС*;

                Вес = *ПеревозВес*;

            }

            public int МД;

            public int Вес;

            public override string База

            {

                get

                {

                    return "Мощьность двигателя: " + МД + " Л.С. " + "максимальный перевозимый вес: " + Вес + "кг " + base.База;

                }

            }

        }

            public enum СтранаПроизводитель

        {

            Россия,

            Германия,

            Америка,

            Англия,

            Италия,

            Нет\_данных

        }

            public enum МаркаТехСред

        {

            Опель,

            Лада,

            Ферари,

            Фольтсваген,

            Нисан,

            Нет\_данных

        }

            public enum Тип

        {

            Седан,

            Универсал,

            Хетчбэк,

            Лифтбэк,

            Пикап,

            Минивэн,

            Нет\_данных

        }

}

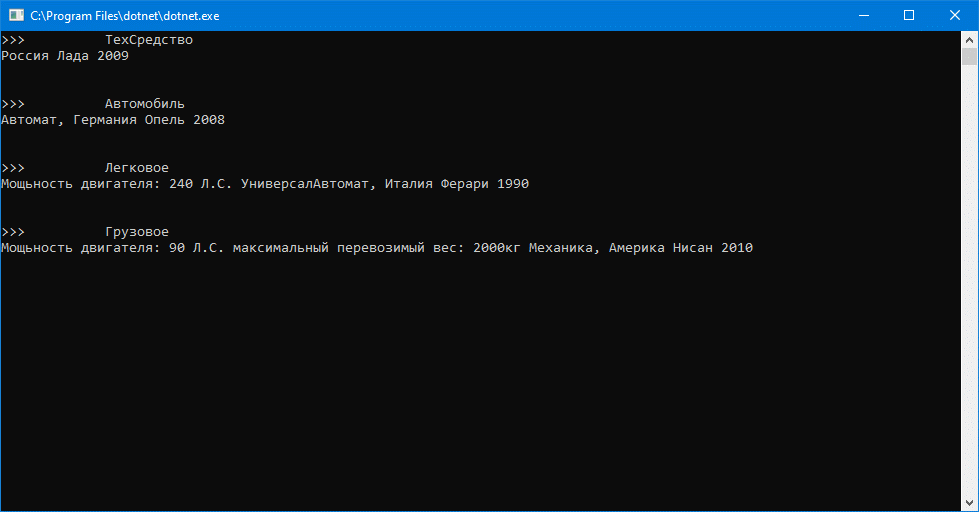
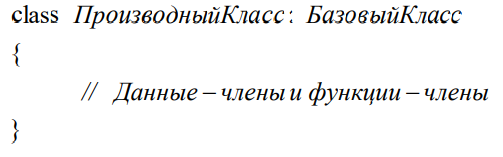


Рисунок 2 – Результат работы программы индивидуального задания

Контрольные вопросы:

1. Что такое наследование реализации? Как описать синтаксически наследование реализации?

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



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

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

2. Для чего используется ключевое слово base?

С помощью ключевого слова base мы можем обратиться к базовому классу.

3. Можно ли переопределить метод класса? Свойства класса? Данные класса?

При наследовании нередко возникает необходимость изменить в классе-наследнике функционал метода, который был унаследован от базового класса. В этом случае класс-наследник может переопределять методы и свойства базового класса.

Те методы и свойства, которые мы хотим сделать доступными для переопределения, в базовом классе помечается модификатором virtual. Такие методы и свойства называют виртуальными.

А чтобы переопределить метод в классе-наследнике, этот метод определяется с модификатором override. Переопределенный метод в классе-наследнике должен иметь тот же набор параметров, что и виртуальный метод в базовом классе.

4. Как переопределить метод в производном классе?

А чтобы переопределить метод в классе-наследнике, этот метод определяется с модификатором override.

5. Для чего используется ключевое слово virtual?

Те методы и свойства, которые мы хотим сделать доступными для переопределения, в базовом классе помечается модификатором virtual. Такие методы и свойства называют виртуальными.

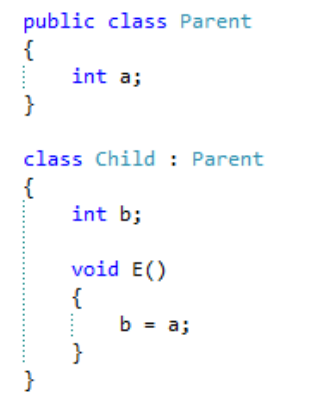
6. Для чего используется ключевое слово override?

А чтобы переопределить метод в классе-наследнике, этот метод определяется с модификатором override.

7. Как поменять цвет фона в консольном приложении?

Console.ForegroundColor = ConsoleColor.Red;

8. Укажите ошибки (если есть) в представленном фрагменте:



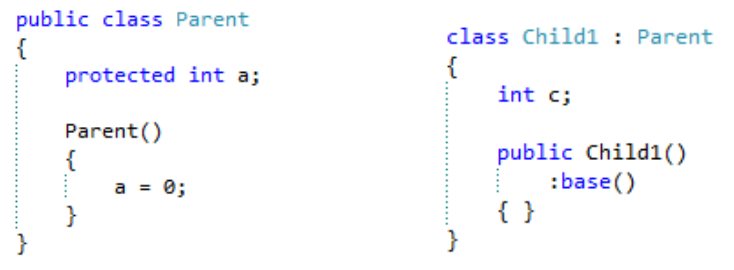
Int a имеет модификатор доступа private

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Серьезность | Код | Описание | Проект | Файл | Строка | Состояние подавления |
| Ошибка | CS0122 | "Program.Parent.a" недоступен из-за его уровня защиты. | L\_8 | C:\Users\kuvsh\Desktop\СКФУ\2\_3\_семестр\Технологии программирования\Programming\_technologies\L\_8\Program.cs | 17 | Активные |

9. Сколько базовых классов может быть у любого класса в C#

Для любого производного класса можно указать только один базовый класс. В C# не предусмотрено наследование нескольких базовых классов в одном производном классе.

10. Даны определения классов на языке C#.



Существуют ли ошибки в данном коде (укажите ошибки, если есть)?

Конструктор Parent() имеет модификатор доступа private

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Серьезность | Код | Описание | Проект | Файл | Строка | Состояние подавления |
| Ошибка | CS0122 | "Program.Parent.Parent()" недоступен из-за его уровня защиты. | L\_8 | C:\Users\kuvsh\Desktop\СКФУ\2\_3\_семестр\Технологии программирования\Programming\_technologies\L\_8\Program.cs | 20 | Активные |