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*)

        {

            Console.BackgroundColor = ConsoleColor.White;

            Console.ForegroundColor = ConsoleColor.Blue;

            Console.Clear();

            Console.Title = "Лабораторная работа №11";

            Human h = new Human("Иванов Иван Иванович", 50);

            h.Plus(5);

            h.Minus(1);

            h.DrawObject();

            Console.WriteLine("\n\n\n");

            Car car = new Car("Hyndai", "ix35", 120);

            car.Plus(25);

            car.Minus(11);

            car.DrawObject();

            Console.ReadKey();

        }

    }

    public interface ICalculate

    {

        void Plus(int *pPlus*);

        void Minus(int *pMinus*);

    }

    public interface IVisual

    {

        string Name {get; set;}

        void DrawObject();

    }

    public class Human : ICalculate, IVisual

    {

        public Human(string *pFIO*, int *pAge*)

        {

            FIO = *pFIO*;

            Age = *pAge*;

        }

        private string FIO;

        private int Age;

        public string Name

        {

            get

            {

                return FIO + " : " + Age.ToString();

            }

            set

            {

                FIO = value;

            }

        }

        public void Plus(int *pPlus*)

        {

            Age += *pPlus*;

        }

        public void Minus(int *pMinus*)

        {

            Age -= *pMinus*;

        }

        public void DrawObject()

        {

            Console.WriteLine

                (

                    "   o   \n" +

                    " ----- \n" +

                    "   | \n" +

                    "  / \\ \n" +

                    " /   \\ \n"

                );

            Console.WriteLine(Name);

        }

    }

    public class Car : ICalculate, IVisual

    {

        public Car(string *pManuFacturer*, string *pModel*, int *pVelocity*)

        {

            Manufacturer = *pManuFacturer*;

            Model = *pModel*;

            Velocity = *pVelocity*;

        }

        private string Manufacturer;

        private string Model;

        private int Velocity;

        public string Name

        {

            get

            {

                return Manufacturer + " - " + Model + " : " + Velocity.ToString() + "km/h";

            }

            set

            {

                Model = value;

            }

        }

        public void Plus(int *pPlus*)

        {

            Velocity += *pPlus*;

        }

        public void Minus(int *pMinus*)

        {

            Velocity -= *pMinus*;

        }

        public void DrawObject()

        {

            Console.WriteLine

                (

                    "       --------------\n"+

                    "\_\_\_\_\_\_/              \\\_\_\_\_\_\n"+

                    "|                           |\n"+

                    "----(@)-----------------(@)--\n"

                );

            Console.WriteLine(Name);

        }

    }

}

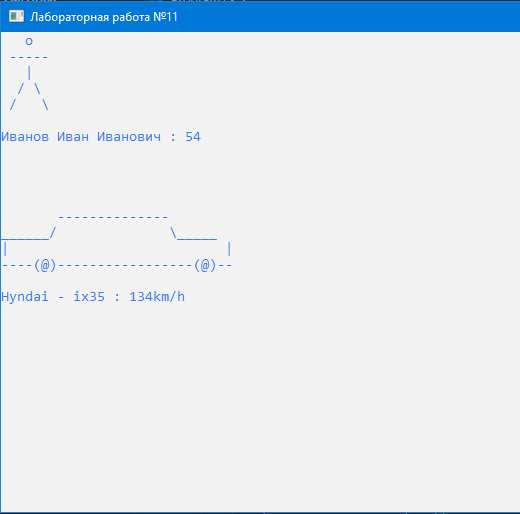


Рисунок 1 – Результат работы программы

**Индивидуальное задание**:

namespace Helloworld

{

    class Program

    {

        static void Main(string[] *args*)

        {

            ТехСредство[] mas = new ТехСредство[5];

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

            mas[0].Add(5);

            mas[0].Drowing();

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

            mas[1].Add(10);

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

            mas[2].Min(1);

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

            mas[4] = new Вертолёт(СтранаПроизводитель.Англия, МаркаТехСред.Фольтсваген, 2022 , 45);

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

            {

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

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

                Console.WriteLine("\n");

            }

            Console.ReadKey();

        }

    }

    public interface ICalc

    {

        void Add (int *pAdd*);

        void Min (int *pMin*);

    }

        public interface Iimg

    {

        void Drowing();

    }

    class ТехСредство : ICalc, Iimg

        {

            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 + "\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";}

            }

        public void Add(int *pAdd*)

        {

            ГодПроизводства += *pAdd*;

        }

        public void Min(int *pMin*)

        {

            ГодПроизводства -= *pMin*;

        }

        public void Drowing()

        {

            Console.WriteLine

            (

               "                       \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_                            \n"+

               "                     %%#/////////////&\*//////////////(@                          \n"+

               "                    %#///////////////@\*////////////////@,                        \n"+

               "                   %%////////////////@\*/////////////////%\*                       \n"+

               "                 %%%////////////////@\*//////////////////%(                       \n"+

               "         ,%%%%%%%%%%%%%%%%%%%%%%%%%%%@%%%%%%%%%%%%%%%%%%%%&%%%%%%%%%%%%%%%%/     \n"+

               "     .%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%@%%%%%%%%%%%%%%%%%%%%&%%%%%%%%%%%%%%%%%%%#  \n"+

               " %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%@%%%%%%%%%%%%%%%%%%%%&%%%%%%%%%%%%%%%%%%%&@    \n"+

               " %%%%%%%%%%%%%%%%@@@@@@%%%%%%%%%%%%@%%%%%%%%%%%%%%%%%%%@%%%%@@@@@@@@@@@%%%%%%%   \n"+

               " @@%%%%%%%%%%%&@@#     @@@@%%%%%%%%%@%%%%%%%%%%%%%%%%%%%%%%@@@@#     @@@@@%%%%&  \n"+

               " @ @%%%%%%%%%%@@@ \*(  ,(  @@@%%%%%%%%@%%%%%%%%%%%%%%%%%%%%%@@@@ \*(  ,(  @@@@%%%% \n"+

               " @ @%%%%%%%%%@@@@ .(. %/  @@@@%%%%%%%%%%%%%%%%%%%%%%%%%%%%%@@@@  (. %/  @@@      \n"+

               "         ( ,@@@@    .@@@@ @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@./@@@@    .@@@@          \n"+

               "                 @@@@@@@@@                                     @@@@@@@@@         \n"

            );

        }

    }

        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;

                }

            }

        public void DrowingLeg()

        {

            throw new NotImplementedException();

        }

    }

        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

        }

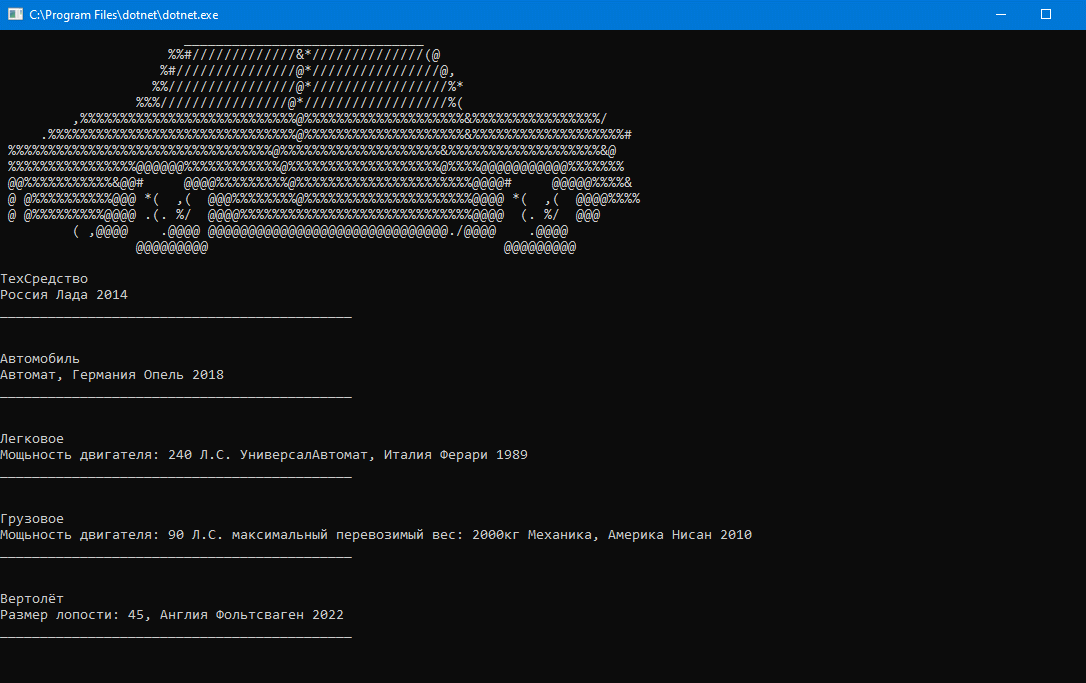


Рисунок 2 – Результат работы программы индивидуального задания

**Контрольные вопросы**

1. Чем отличается наследование интерфейсов от наследования реализации?

Наследование реализации – это и есть наследование.

Наследование интерфейса – это реализация интерфейса или абстрактных методов абстрактного класса.

2. Поясните, каким образом проявляется полиморфное поведение объектов при реализации классами интерфейсов?