**Лабораторная работа 11 Часть 2**

*«Классы»*

**Рекомендации по программированию**

* Поля обязательно делать private.
* Интерфейс должен быть полным, т.е. предоставлять возможность выполнять любые разумные действия с классом. И одновременно минимально необходимым, т.е. без дублирования и пересечения возможностей методов.
* Каждый метод класса должен решать только одну задачу. Не стоит объединять два коротких независимых фрагмента кода в один метод.
* Если действия встречаются хотя бы дважды, стоит оформить их в отдельные функции.
* Все заданные методы, поля и экземпляры классов должны иметь осмысленные имена.

**Требования к лабораторной**

* Каждый разрабатываемый класс должен, как правило, содержать следующие элементы: поля с различными спецификаторами, конструкторы с параметрами и без параметров, методы, свойства.
* Методы и свойства должны обеспечивать непротиворечивый, полный, минимальный и удобный интерфейс класса.

**Постановка задачи**

1. Создать класс согласно вашему индивидуальному заданию. Объявить **не менее пяти** полей-характеристик для данного класса, используя различные спецификаторы доступа. Реализовать механизмы **ввода и вывода** данных для одного объекта этого класса.

*Пример: Класс Псайдак с полями: имя, пол, вес, рост, цвет, сила, физмастерство, уклонение, лечение, арсенал, боевая способность (разгромный удар), не боевая способность (телекинез) и другие.*

1. Для предыдущего задания реализовать все возможные типы **конструкторов** (не менее трех непустых). Реализовать **свойства**, позволяющие осуществить проверку на допустимые значения для данных класса. Создать **массив** объектов данного класса. Каждый объект должен быть уникальным. Вывести этот массив на консоль.
2. Для предыдущего задания реализовать **не менее шести** **полезных** методов для работы с заданным классом. Также реализовать **методы сортировки по каждому имеющемуся полю**, **вывода максимального и минимального значения поля** (где таковое возможно).

*Например, поиск оптимального удара среди ударов ногой, рукой, головой (максимальный урон от удара, который придется соперника) у каждого персонажа этой расы. Методы для расчета ударов ногой, рукой и головой должны быть отдельные и свои.*

|  |  |
| --- | --- |
| **№** | **Класс** |
| 1 | Автомобиль |
| 2 | Покемон |
| 3 | Греческий Бог |
| 4 | Фрукт |
| 5 | Персонаж игры Heroes Might and Magic |
| 6 | Сказочное существо |
| 7 | Компьютер |
| 8 | Работник |
| 9 | Мебель |
| 10 | Напиток |
| 11 | Песня |
| 12 | Транспорт |
| 13 | Гриб |
| 14 | Овощ |
| 15 | Еда |
| 16 | Человек |
| 17 | Птица |
| 18 | Дверь |
| 19 | Сущность во вселенной Лавкрафта |
| 20 | Фильм |
| 21 | Книга |
| 22 | Здание |
| 23 | Насекомое |
| 24 | Окно |
| 25 | Персонаж игры Witcher |
| 26 | Млекопитающее |
| 27 | Фентези существо |
| 28 | Персонаж игры Assassin’s Creed |

class Car

{

public string mark;

public int doors;

public double price;

public double speed;

public double probeg;

public int bakVolum;

public double Price { get { return price; } }

public Car(double price, double speed, double probeg, string mark, int doors, int bakVolum)

{

this.mark = mark;

this.price = price;

this.probeg = probeg;

this.speed = speed;

this.doors = doors;

this.bakVolum = bakVolum;

}

public Car(double price, string mark, int doors)

{

this.mark = mark;

this.price = price;

this.probeg = 0;

this.speed = 0;

this.doors = doors;

}

public Car()

{

this.mark = "";

this.price = 0;

this.probeg = 0;

this.speed = 0;

this.doors = 0;

}

public Car(double price)

{

this.mark = "";

this.price = price;

this.probeg = 0;

this.speed = 0;

this.doors = 0;

}

public void InputInfo()

{

this.mark = Console.ReadLine();

this.price = Convert.ToDouble(Console.ReadLine());

this.probeg = Convert.ToDouble(Console.ReadLine());

this.speed = Convert.ToDouble(Console.ReadLine());

this.doors = Convert.ToInt32(Console.ReadLine());

this.bakVolum = Convert.ToInt32(Console.ReadLine());

}

public double Time(int range)

{

return range / speed \* 60;

}

public double Fuel(int range, int fuel)

{

return range / fuel;

}

public double NewProbeg(int range)

{

return probeg + range;

}

public int Maslo(int range)

{

return range / 3000;

}

public int BakVolume(int range)

{

if (bakVolum==0)

{

return 0;

}

return range / bakVolum;

}

public int Omivaika(int range, int omivayka)

{

return range / omivayka;

}

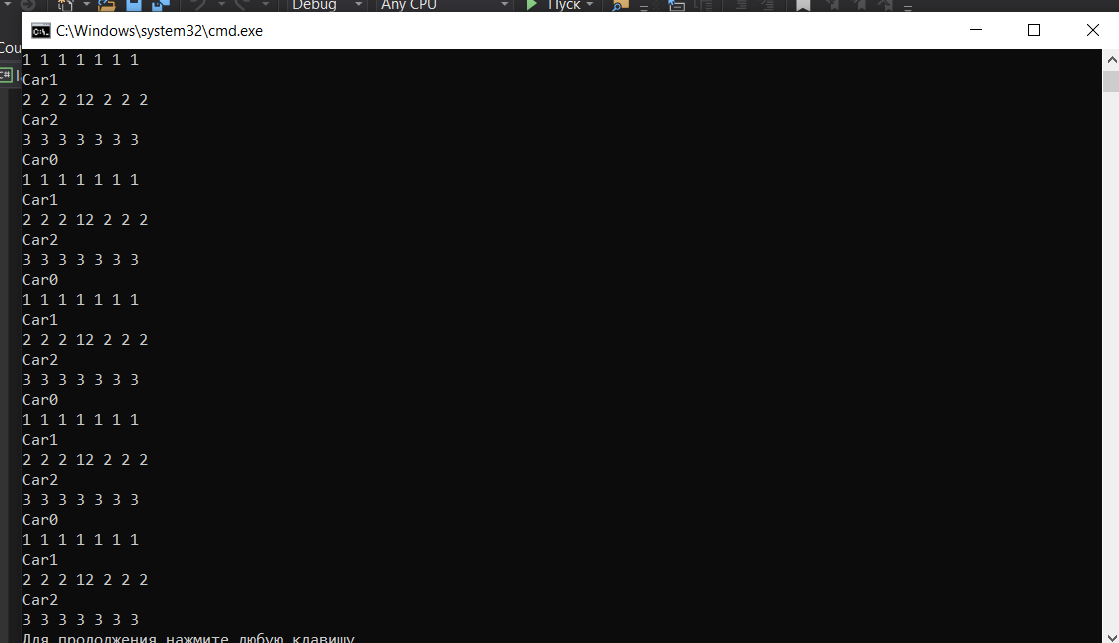
public void ShowInfo()

{

Console.WriteLine(price + " " + speed + " " + probeg + " " + mark + " " + doors + " " + bakVolum + " " + price);

}

}

****