

Лабораторная работа №9

Тема: Создание и применение структур и перечислений

Цель работы: Формирование умений и навыков в разработке и типов данных структуры.

Время на выполнение работы: 2 часа

Этапы работы:

I. Выполнить задания.

I. *Выполнение задания*

Пояснения:

Необходимо объявить структуру в отдельном файле, так же в отдельном файле объявить перечисление и его элементы (3-7 штук). В файле с основным кодом в main создать массив структур. Описать методы для работы с ними. При запуске программы выводить меню и выполнять действия, указанные в нем.

Ввести массив структур в соответствии с вариантом. В программе реализовать меню:

- 1) Ввод массива структур;
- 2) Изменение заданной структуры по id;
- 3) Вывод на экран массива структур;
- 4) Специальный пункт 1
- 5) Специальный пункт 2
- 6) Выход.

Пример: Структура «Автосервис»: регистрационный номер автомобиля, марка(перечисление), пробег, мастер, выполнивший ремонт, сумма ремонта.

Вывести общий пробег по всем машинам одной марки

Вывести общую сумму ремонта по каждому мастеру

```
enum Brend
{
    opel,
    bmw,
    honda,
    mercedes,
    lada,
    kia,
    landrover
}

struct Autoservice
{
    public string regNomer;
    public Brend breedCar;
    public int odometr;
    public string surnameMaster;
    public double price;
}
```

```

static void Main(string[] args)
{
    while (true)
    {
        Menu.MenuText();
        int item = int.Parse(Console.ReadLine());
        Menu.SelectMenuItem(item);
    }
}

public static void MenuText()
{
    Console.WriteLine("1 - Ввод массива структур");
    Console.WriteLine("2 - изменение структуры структуры");
    Console.WriteLine("3 - Вывод на экран массива структур");
    Console.WriteLine("4 - специальный пункт1 (общий пробег по одной марке)");
    Console.WriteLine("5 - специальный пункт2 (общая сумма ремонта по одному мастеру)");
    Console.WriteLine("6 - Выход");
    Console.WriteLine("-----");
}

public static void SelectMenuItem(int item)
{
    switch (item)
    {
        case 1: WorkStruct.CreateArrayStructAutoservice();break;
        case 2: WorkStruct.UpdateStructFromId();break;
        case 3: WorkStruct.PrintStruct();break;
        case 4: Console.WriteLine(WorkStruct.AmountAdometrCurrentBrendCar());break;
        case 5: Console.WriteLine(WorkStruct.AmountPriceCurrentMaster());break;
        case 6: Environment.Exit(0);break;
        default: Console.WriteLine("такого пункта нет");break;
    }
}

class WorkStruct
{
    static Autoservice[] autoservices;

    ссылка: 1
    public static void CreateArrayStructAutoservice()
    {
        Console.WriteLine("Сколько структур создать?");
        int count = int.Parse(Console.ReadLine());
        autoservices = new Autoservice[count];
        WriteData();
    }

    private static void WriteData()
    {
        for (int i = 0; i < autoservices.Length; i++)
        {
            Console.WriteLine("введите гос номер авто");
            autoservices[i].regNomer = Console.ReadLine();
            Console.WriteLine("выберите марку авто. для выбора укажите номер марки");
            PrintBrendCar();
            autoservices[i].brendCar = (Brend)int.Parse(Console.ReadLine());

            Console.WriteLine("введите пробег в целых километрах");
            autoservices[i].odometr = int.Parse(Console.ReadLine());

            Console.WriteLine("Введите фамилию мастера");
            autoservices[i].surnameMaster = Console.ReadLine();

            Console.WriteLine("Введите конечную стоимость ремонта");
            autoservices[i].price = double.Parse(Console.ReadLine());

            Console.WriteLine("-----");
        }
    }
}

```

Ссылка: 3

```
private static void PrintBrendCar()
{
    int j = 0;
    foreach (var item in Enum.GetNames(typeof(Brend)))
    {
        Console.WriteLine($"{j++} - {item}");
    }
}
```

Ссылка: 1

```
public static void PrintStruct()
{
    for (int i = 0; i < autoservices.Length; i++)
    {
        Console.WriteLine($"{i + 1}|{autoservices[i].regNomer}|{autoservices[i].brendCar}|" +
            $"{autoservices[i].odometr}|{autoservices[i].surnameMaster}|{autoservices[i].price}");
    }
}

public static int AmountAdometrCurrentBrendCar()
{
    Console.WriteLine("выберите марку автомобиля по которой вывести километраж. укажите номер");
    PrintBrendCar();
    int brend = int.Parse(Console.ReadLine());
    int amountAdometr = 0;
    for (int i = 0; i < autoservices.Length; i++)
    {
        if (autoservices[i].brendCar == (Brend)brend)
            amountAdometr += autoservices[i].odometr;
    }
    return amountAdometr;
}
```

Ссылка: 1

```
public static double AmountPriceCurrentMaster()
{
    Console.WriteLine("напишите фамилию мастера");
    string surname = Console.ReadLine();
    double amountPrice = 0;
    for (int i = 0; i < autoservices.Length; i++)
    {
        if (autoservices[i].surnameMaster == surname)
            amountPrice += autoservices[i].price;
    }
    return amountPrice;
}

public static void UpdateStructFromId()
{
    Console.WriteLine("укажите номер в который хотите внести изменения");
    int n = int.Parse(Console.ReadLine());
    for (int i = 0; i < autoservices.Length; i++)
    {
        if (i == n - 1)
        {
            Update(i);
        }
    }
}
```

```

private static void Update(int i)
{
    Console.WriteLine("хотите поменять номер? y/n");
    if (CheckChoise())
        autoservices[i].regNomer = Console.ReadLine();

    Console.WriteLine("хотите поменять марку? y/n");
    if (CheckChoise())
    {
        PrintBrendCar();
        autoservices[i].brendCar = (Brend)int.Parse(Console.ReadLine());
    }

    Console.WriteLine("хотите поменять пробег? y/n");
    if (CheckChoise())
        autoservices[i].odometr = int.Parse(Console.ReadLine());

    Console.WriteLine("хотите поменять фамилию мастера? y/n");
    if (CheckChoise())
        autoservices[i].surnameMaster = Console.ReadLine();

    Console.WriteLine("хотите поменять стоимость? y/n");
    if (CheckChoise())
        autoservices[i].price = double.Parse(Console.ReadLine());
}

public static bool CheckChoise()
{
    return Console.ReadLine() == "y" ? true : false;
}

```

Варианты:

1. Структура «Сотрудник»: фамилия, имя, отчество; должность(перечисление); год рождения; заработная плата.
Вывести количество однофамильцев по каждой фамилии
Вывести общую заработную плату по каждой должности
2. Структура «Государство»: форма правления(перечисления); название; столица; численность населения; занимаемая площадь.
Вывести общую численность населения по всем формам правления
Вывести общую занимаемую площадь по всем формам правления
3. Структура «Человек»: фамилия, имя, отчество; домашний адрес; номер телефона, оператор(перечисление), возраст.
Вывести количество людей с одинаковым именем для каждого имени
Посчитать общий возраст людей, зарегистрированных у одного оператора
4. Структура «Читатель»: Фамилия И.О., номер читательского билета, название книги, жанр(перечисление), срок возврата.
Вывести все книги срок возврата которых истек
Посчитать количество книг по всем жанрам
5. Структура «Школьник»: фамилия, имя, отчество; класс№; классБуква(перечисление); оценки по предметам (математика, физика, русский язык, литература).
Посчитать количество студентов по каждому отчеству
Посчитать средний бал по каждому классу

6. Структура «Студент»: фамилия, имя, отчество; домашний адрес; группа; рейтинг.
Посчитать средний рейтинг по группам
Посчитать количество студентов по группам
7. Структура «Покупатель»: фамилия, имя, отчество; домашний адрес; номер телефона; номер кредитной карточки; вид карточки(перечисление)
Вывести количество людей с одинаковым именем для каждого имени
Посчитать сумму цифр кредитной карты для каждого пользователя
8. Структура «Пациент»: фамилия, имя, отчество; домашний адрес; номер медицинской карты; номер страхового полиса. (2 буквы(сделать перечисление) и 6 цифр)
Вывести количество однофамильцев по каждой фамилии
Вывести сумму цифр страхового полиса для каждой двух букв по всем клиентам
9. Структура «Информация»: носитель(перечисление); объем; название; автор.
Вывести общий объем информации для каждого автора
Вывести общий объем информации для каждого вида носителя
10. Структура «Клиент банка»: Фамилия И.О., Банк(перечисление); номер счета, сумма на счете, дата последнего изменения.
Вывести общую сумму вкладов для каждого банка
Вывести общую сумму приходов за сегодня для каждого банка
11. Структура «Склад»: наименование товара, цена, количество, процент торговой надбавки.
Посчитать общую выгоду по каждому наименованию товара(количество*цена*(100+%торговой надбавки))
Посчитать общее количество товара, у которого стоимость ниже 100 BYN
12. Структура «Авиарейсы»: номер рейса, пункт назначения(перечисление), время вылета, дата вылета, стоимость билета.
Посчитать общее время полета по всем пунктам назначения
Посчитать общую стоимость всех билетов по всем пунктам назначения
13. Структура «Вокзал»: номер поезда, пункт назначения, дни следования(перечисление), время прибытия, время стоянки.
Посчитать общее время простоя поездов во все пункты назначения
Посчитать сумму цифр всех поездов, следующих в определенные дни и вывести для всех.

Литература

1. Полный справочник по С#. Г. Шилдт. Издательский дом «Вильямс», 2004.
2. С# в подлиннике. Наиболее полное руководство. Х.Дейтел.
3. С# в задачах и примерах. Культин. Н.Б.
4. С# учебный курс. Г.Шилдт. СПб.: Питер, 2002.
5. С# программирование на языке высокого уровня Павловская Т.А. СПб.: БХВ-Петербург.