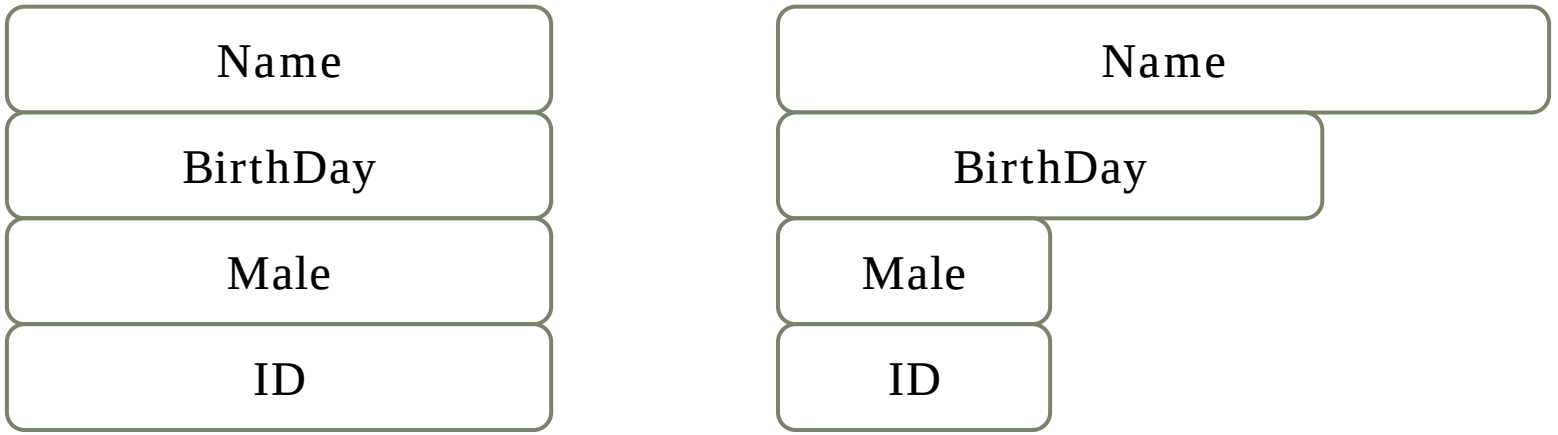
Лекция 15. Структуры

Курс «Программирование»
ПрИн+МОиАИС, 2 семестр

Щукин Александр
Валентинович

Необходимость и понятие

- Стандартные типы данных и их ограниченность
- Структура – это тип
- Структура - a struct type is a value type that is typically used to encapsulate small groups of related variables, such as the coordinates of a rectangle or the characteristics of an item in an inventory.



Классика и CS

- В ряде классических ЯП структура может содержать только поля (fields)
- В C#: Structs can also contain constructors, constants, fields, methods, properties, indexers, operators, events, and nested types, although if several such members are required, you should consider making your type a class instead.
- «Облегченные классы»

Пример объявления

```
// точка на плоскости
public struct Point
{
    public int X;
    public int Y;
}
Point p;
p.X = 100;
p.Y = -50;
```

```
// студент
public struct Student
{
  public string Name;
  DateTime BDay;
  long ID;
  string GroupID;
}
Student s;
S.ID = 1234567;
S.GroupID = “3530203.90001”
```

Объявление структур и переменных

- Структура объявляется на уровне модуля (класса)
- Переменная объявляется как обычно, но ее тип - структура
- Структура – value type

Область видимости

- Область видимости
  - Структуры – часть кода, в которой можно обращаться к структуре
  - Члены структуры – возможность обращения извне
  - Переменных типа структура
- Есть ограничения. Например, переменная не может быть public, если структура private

Конструктор

- Инициализация полей – явная и неявная

```
// точка на плоскости
public struct Point
{
    public int X;
    public int Y;
    public Point(int NewX, int NewY)
    {
        X = NewX;
        Y = NewY;
    }
}
Point p;
p = new Point();      //используется конструктор по умолчанию
p = new Point(20,30);
```

Методы

```
// точка на плоскости
public struct Point
{
    public int X;
    public int Y;
    public Point(int NewX, int NewY)
    {
       X = NewX;
       Y = NewY;
    }
    public void Increment()
    { X++; Y++; }
    public void Decrement()
    { X--; Y--; }
    public void Display()
    { Console.WriteLine("X= {0}; Y= {1}", X, Y); }
}
Point p = new Point(20,30);
p.Increment();
p.Display();
```

Особые случаи. Структура компьютер

```
public struct Comp
{
    public int RAM;
    public string Proc;
    public float Price;
    public int HDD;
}
```

Особые случаи. Поле - массив

```
public struct Comp
{
    public int RAM;
    public string Proc;
    public float Price;
    public int[] Drives;
}
Comp z;
z.Drives = new int[3];
z.Drives[0] = 100;
```

Особые случаи. Массив структур

```
// массив компьютеров
Comp[] officeComps = Comp[100];
officeComps[15].Drives = new int[3];
officeComps[15].Drives[0] = 100;
```

Особые случаи. Вложенные структуры

```
public struct Drive
{
    public int Size;
    public string DriveType;
}
public struct Comp
{
    public int RAM;
    public string Proc;
    public float Price;
    public Drive[] Drives;
}
Comp z;
z.Drives = new Drive[3];
z.Drives[0].Size = 100;
```

Присвоение структур

```
public struct Point1
{
    public int X;
    public int Y;
}
Point1 p1;
public struct Point2
{
    public int X;
    public int Y;
}
Point2 p2;
p2 = p1;                // ТАК НЕЛЬЗЯ!
```

Ссылочный тип и тип значения

	Тип значения	Ссылочный тип
Где размещается	В стеке	В управляемой куче
Как представляется переменная	В виде локальной копии	В виде ссылки, указывающей на занимаемое соответствующим экземпляром место в памяти
Какой тип является базовым	Наследуется от System.ValueType	Может от любого другого типа (не System.ValueType)
Может ли тип выступать в роли базового (наследование)	Нет	Да
Может ли иметь конструктор	Да, один есть предопределенный	Да, все конструкторы нужно создавать
Когда переменные уничтожаются	Когда выходят за область видимости	При сборке мусора

Правила хорошего тона

- Грамотное проектирование структуры
- Не объявляйте зависимых полей. Вместо этого используйте методы
- Единицы измерения