



C# Basic



Проверить, идет ли запись

Меня хорошо видно && слышно?



Правила вебинара



Активно
участвуем



Off-topic обсуждаем
в Telegram
#csharp-basic-2024-05



Задаем вопрос
в чат или голосом



Вопросы вижу в чате,
могу ответить не сразу

Условные обозначения



Индивидуально



Время, необходимое
на активность



Пишем в чат



Говорим голосом



Документ



Ответьте себе или
задайте вопрос

Структуры и перечислени

Цели вебинара

После занятия вы сможете:

1. Пользоваться структурами и понимать принцип их работы
2. Использовать исключения для построения доступного для понимания кода
3. Работать с операторами булевой алгебры

Преподаватель



Пилипсон Эдгар

Senior Software Engineer, Luxoft



План полета



Что такое структура, виды

Перечисления

Битовые операции

Кодинг

Рефлексия

Структуры

Что такое

Структура - значимый тип, который инкапсулирует данные и функциональность.

Размещаются в стеке

Примитивные типы (int, float, bool...) - структуры

Чтобы объявить структуру используется ключевое слово struct

```
struct UserInfo
{
    public string Name;
    public byte Age;

    public void WriteUserInfo()
    {
        Console.WriteLine("Имя: {0}, возраст: {1}", Name, Age);
    }
}
```



Зачем

цель: повысить производительность приложения

работает для небольших структур

Зачем

- нельзя определить конструктор по умолчанию (до C#10)
- не поддерживает наследования
- => `abstract`, `virtual` или `protected` - не указываются у членов структур
- нельзя определить деструктор
- передается по значению: при присвоении, передаче параметра в метод или возвращаемое значение метода значение копируется
- `Equals()`: сравнение по значениям

Перечисления

Что такое

Перечисление - целочисленный тип, определенный набором именованных констант

Удобен для хранения возможных значений области (дни недели, месяцы, поддерживаемые типы устройств и пр.) , состояния сущности (соединения с БД, код http ответа и пр.)

Чтобы объявить перечисление используется ключевое слово **enum**

```
enum Season
```

```
{
```

```
    Spring, // имеет
```

```
    Summer,
```

```
    Autumn,
```

```
    Winter
```

```
}
```

Чем хороши

- Облегчает сопровождение
- Делают код яснее
- Ускорение написания кода

Булева алгебра

Или / or

в булевой алгебре:

Дизъюнкция / логическое сложение

обозначение \vee , $|$

ложно тогда, когда тогда и только тогда,

когда ложны оба высказывания

(аргумента)

запись в c# (и почти везде):

`int c = a | b`

пример побитового сложения:

A	B	A B
0	0	0
1	0	1
0	1	1
1	1	1

И / and

в булевой алгебре:

Конъюнкция / логическое
умножение

обозначение \wedge , &

ИСТИННО тогда, когда *тогда* и
ТОЛЬКО тогда, когда *ИСТИННЫ* оба
высказывания

запись в с# (и почти везде):

```
int c = a & b
```

A	B	A&B
0	0	0
1	0	0
0	1	0
1	1	1

Не

в булевой алгебре:

Инверсия, отрицание

обозначение \neg

результатом является
суждение, «противоположное»
исходному

запись в C# (и почти везде):

`bool c = !a`

A	$\neg A$
0	1
1	0

Флаги



Битовые флаги

Если необходимо, чтобы значение перечисления представлял комбинацию вариантов:

- определить каждое значение как степень двойки
- использовать побитовые логические операторы И / ИЛИ

На практике можно применять и к перечислениям

Булева алгебра

XOR

в булевой алгебре:

Исключающее ИЛИ, сложение
по модулю 2

обозначение **XOR**, \oplus

истинно тогда и только тогда,
когда один из аргументов истинен,
а другой — ложен

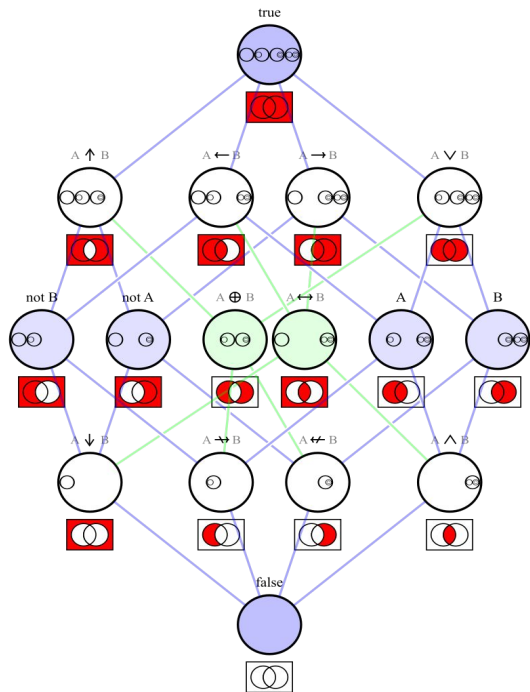
запись в c# (и почти везде):

```
int c = a ^ b
```

пример побитового XOR:

A	B	A^B
0	0	0
1	0	1
0	1	1
1	1	0

Еще материалы



можно смотреть в википедии

- Дизъюнкция
- Импликация
- Конъюнкция
- Отрицание (унарная)
- Исключающее «или»
- Стрелка Пирса
- Условная дизъюнкция (тернарная)
- Штрих Шеффера
- Эквиваленция

Выводы и рефлексия

Цели вебинара

Тезисы

1. Структуры позволяют в определенных сценариях сэкономить ресурсы
2. Перечисления удобный инструмент для работы с ограниченным количеством значений



Рефлексия



По какому вопросу захотелось глубже изучить информацию?



Понимаете ли вы, как применять на практике то, что узнали на вебинаре. Если да, то как?

Следующий вебинар



26 июля 2024

Модуль 2: ООП и анонимные типы с методами

Групповая менторская консультация



Ссылка на вебинар
будет в ЛК за 15 минут



Материалы
к занятию в ЛК —
можно изучать



Обязательный материал
обозначен красной
лентой



**Заполните, пожалуйста,
опрос**

Спасибо за внимание!
Приходите на следующие
вебинары