

## РК1

**Вопрос: Как работает оператор сопоставления с образцом на основе оператора switch в языке C#?**

**Ответ:**

Для выполнения действий в switch используются односторонние выражения. Это позволяет конструкции сохранить легкость для чтения и понимания. Если в каком-то случае необходимо выполнить более одного действия, то их можно записать в виде локального метода или класса.

```
public static int Count<T>(this IEnumerable<T> e)
{
    switch (e)
    {
        case ICollection<T> c: return c.Count;
        case IReadOnlyCollection<T> c: return c.Count;
        // Matches concurrent collections
        case IProducerConsumerCollection<T> pc: return pc.Count;
        // Matches if e is not null
        case IEnumerable<T> _: return e.Count();
        // Default case is handled when e is null
        default: return 0;
    }
}
```

В этом примере показан первый набор операций оператора switch:

1. С оператором switch может использоваться переменная любого типа.
2. Предложением case можно задать образец.
3. Важен порядок предложений case. Компилятор выдаст ошибку, если предыдущее предложение соответствует базовому типу, а последующее — производному.
4. Нестандартные предложения неявно проверяются на значение null\*\*. В приведенном выше примере последнее предложение case является действительным, так как совпадает, только когда аргумент не равен null.

В последнем предложении case показана еще одна функция, добавленная в C# 7, — образцы пустой переменной. Специальное имя \_ сообщает компилятору, что переменная не нужна. Образцу типа в предложении case требуется псевдоним. Но если вам это не нужно, можно использовать \_.