

# Лекция 8. Область видимости и время жизни

Курс «Программирование»

КИТ, 1 семестр

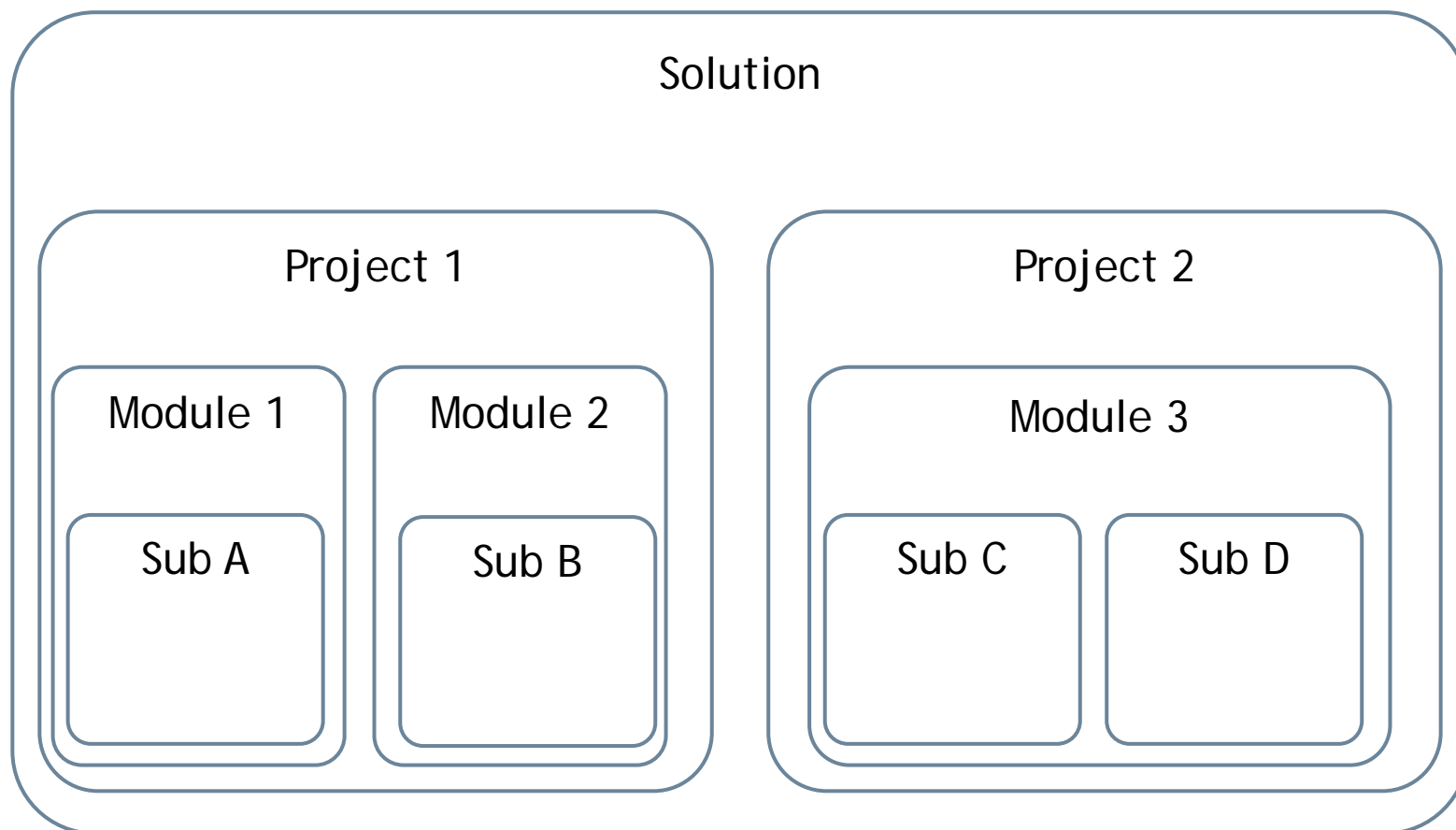
Щукин Александр

Валентинович

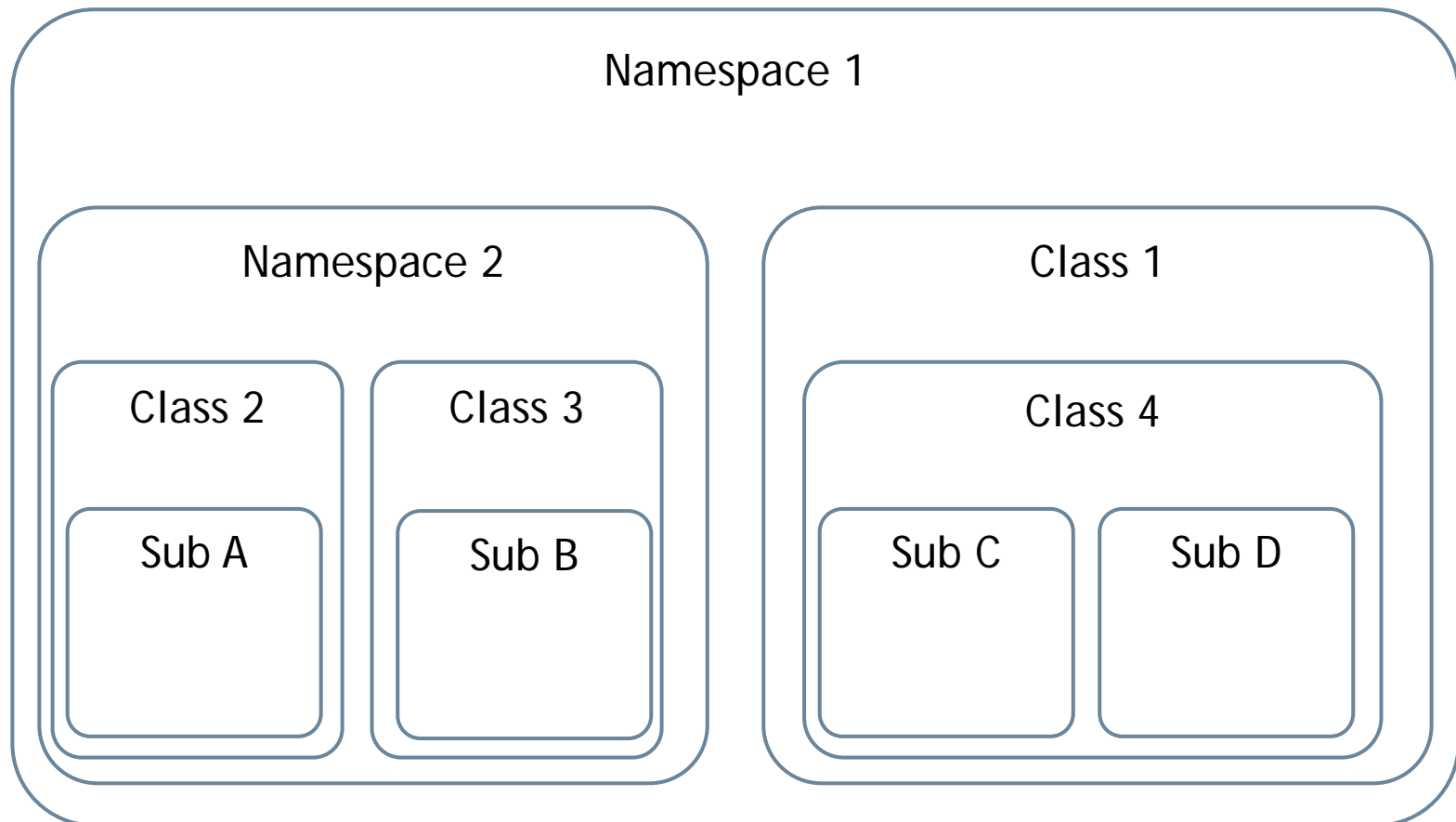
# Область видимости (scope)

- Понятие области видимости в процедурных языках программирования
- Понятие уровней доступности (Accessibility Levels) в C#
- Это может быть использовано для:
  - Переменной
  - Подпрограммы
  - Структуры
  - Структуры

# Процедурный подход



# Подход .Net



# Namespace

- Иерархическая и модульная организация кода
- Оператор «.» используется для разделения
- Директива `using`

# Пример кода

```
namespace SampleNamespace
{
    class SampleClass
    {
        public void SampleMethod()
        {
            System.Console.WriteLine(
                "SampleMethod inside SampleNamespace");
        }
    }
}
```

- Другой пример кода

# Правила обращения

```
namespace N1      // N1
{
    class C1      // N1.C1
    {
        class C2  // N1.C1.C2
        {
        }
    }
    namespace N2  // N1.N2
    {
        class C2  // N1.N2.C2
        {
        }
    }
}
```

# Проблема затенения имен

- Когда не работает: `Console.WriteLine(..)` и даже `System.Console.WriteLine(..)`

```
class TestApp      Global::System.Console.WriteLine(..)
{
    // Define a new class called 'System' to cause problems.
    public class System { }

    // Define a constant called 'Console' to cause more problems.
    const int Console = 7;
    const int number = 66;

    static void Main()
    {
        // The following line causes an error. It accesses TestApp.Console,
        // which is a constant.
        //Console.WriteLine(number);
    }
}
```



# Модификаторы доступа

Access Modifiers	Назначение
Public	Неограниченный доступ
Protected	Класс или типы, производные от этого класса
Internal	Текущая сборка
Protected internal	2) + 3)
Private	Содержащий тип

Результирующая область видимости зависит от уровня доступности и домена доступности. См. пример:

[https://msdn.microsoft.com/ru-ru/library/zd74a9ww\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/zd74a9ww(v=vs.110).aspx)

# Модификатор `static`

- Используется для предоставления доступа к члену (переменная, подпрограмма) через тип (модуль), а не через объект

# Лекция 8. Константы и перечисления

Курс «Программирование»

КИТ, 1 семестр

Щукин Александр

Валентинович

# Назначение

- Читабельность программы
- Вероятность опечатки
- Централизованное изменение значения
- Константы из известных библиотек
  - `System.Math.Pi` `System.Math.E`
  - `System.ConsoleColor.Green`
  - `System.ConsoleKey.F1`

# Объявление констант

- `const double Pi = 3.1415926;`
- `public const double TaxAllocation = 0.13;`
- `private const string ErrMsg = "Invalid data";`
- `const int TotalHours = 356*24;`
- `const double DoublePi = Pi * 2;`
- Выражение, которым инициализируем константу должно полностью вычисляться при компиляции. Поэтому там запрещено использовать...

# Перечисление (enum)

- Перечисление используется для объявления набора констант
- `enum Days { Sat, Sun, Mon, Tue, Wed, Thu, Fri };`
- `Console.WriteLine((int) Days.Sat); // 0`
- `private enum Days: short { Sat=1, Sun, Mon, Tue, Wed, Thu, Fri };`

# Примеры использования

- `Days d;`
- `d = Days.Mon;`
- `if (d >= Days.Mon & d <= Days.Fri)`
  
- `short d;`
- `d = (short) Days.Mon;`
- `if (d >= (short) Days.Mon & d <= (short) Days.Fri)`