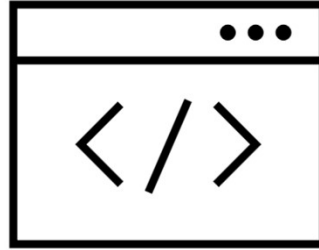


ООП

Инкапсуляция

Наследование

Полиморфизм



Классы. Данные



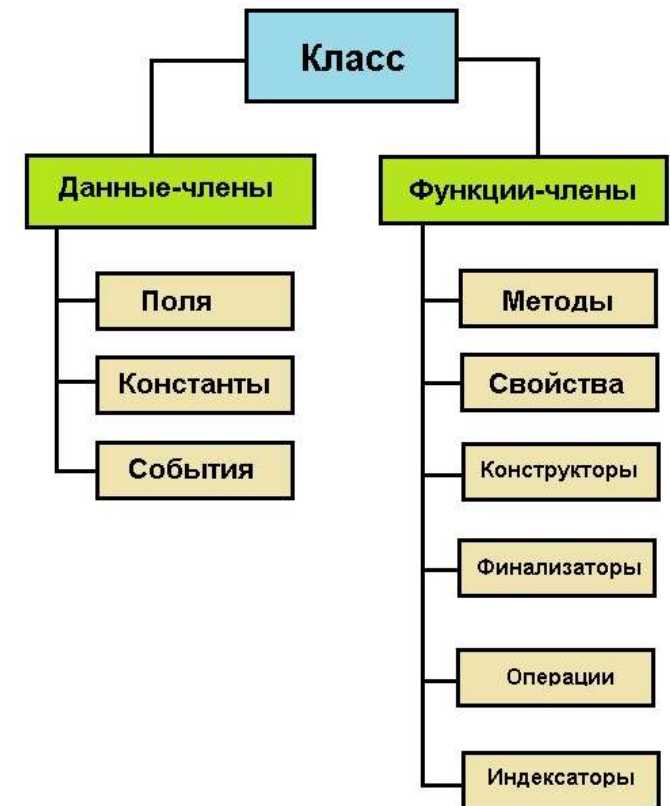
КубГТУ, Кафедра «Информационных систем и программирования»

Данные-члены — это те члены, которые содержат данные класса — поля, константы, события.

Поля (field) – это любые переменные, ассоциированные с классом.

Константа объявляется с помощью ключевого слова `const`. Если она объявлена как `public`, то в этом случае становится доступной извне класса.

События – это члены класса, позволяющие объекту уведомлять вызывающий код о том, что случилось нечто достойное упоминания, например, изменение свойства класса либо некоторое взаимодействие с пользователем.



Синтаксис описания членов-данных:

[атрибуты] [спецификаторы] [const] тип имя [= начальное_значение];

Возможные спецификаторы для данных приведены в таблице

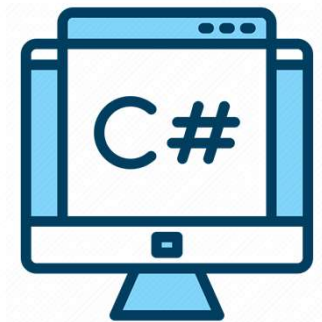
№ п/п	Спецификатор	Описание
1.	internal	Доступ только из данной сборки
2.	new	Новое описание поля, скрывающее унаследованный элемент класса
3.	private	Доступ только из данного класса
4.	protected	Доступ только из данного и производных классов
5.	protected internal	Доступ только из данного и производных классов из данной сборки
6.	public	Доступ к элементу не ограничен
7.	readonly	Поле доступно только для чтения (значения таких полей можно установить либо при описании, либо в конструкторе)
8.	static	Одно поле для всех экземпляров класса
9.	volatile	Поле может изменяться другим процессом или системой



КубГТУ, Кафедра «Информационных систем и программирования»

Для **констант** можно использовать только спецификаторы 1–6.

По умолчанию члены-данные считаются закрытыми, т.е. для них установлен спецификатор **private**. Для членов-данных этот вид доступа является предпочтительным, поскольку поля определяют внутреннее строение класса, которое должно быть скрыто от пользователя. Все методы класса имеют непосредственный доступ к его закрытым полям.



Спецификаторы доступа позволяют задать допустимую область видимости для компонентов класса. То есть спецификаторы доступа определяют контекст, в котором можно употреблять данную переменную или метод.



КубГТУ, Кафедра «Информационных систем и программирования»



Поля, описанные со спецификатором **static**, а также константы существуют в единственном экземпляре для всех объектов класса, поэтому к ним обращаются не через имя экземпляра, а через имя класса.

Обращение к полю класса выполняется с помощью операции доступа (точка). Справа от точки задается имя поля, слева – имя экземпляра для обычных полей или имя класса для статических: **объект.поле_класса**



Далее будет рассмотрена программа, включающая в себя класс Circle, который описывает окружность (ее координаты, радиус и т.д.).

```
using System;

namespace MyProgram
{
    class Circle
    {
        public int x=0;
        public int y=0;
        public int radius=3;
        public const double pi = 3.14;
        public static readonly string name = "Окружность";
        double area;
    }
}
```



```
class Program
{
    static void Main()
    {
        Circle oneCircle = new Circle(); // Создание экземпляра класса
        Console.WriteLine("pi={0}", Circle.pi); // Обращение к константе
        // Обращение к статическому полю
        Console.WriteLine("Используется объект {0}", Circle.name);
        // Обращение к обычным полям
        Console.WriteLine("Центр в точке ({0},{1}), радиус {2}", oneCircle.x, oneCircle.y,
oneCircle.radius);
        oneCircle.radius = 100;
        Console.WriteLine(" Новая окружность с центром в точке ({0},{1}) и радиусом
{2}", oneCircle.x, oneCircle.y, oneCircle.radius);
    }
}
```




```
// Так как поле area имеет спецификатор private, то обращение к  
// нему, наподобие следующего, вызовет ошибку:  
// oneCircle.area = 2 * Circle.pi * oneCircle.radius;  
// Попытка изменить значение поля name также вызовет ошибку, т.к.  
// это поле доступно только для чтения:  
// Circle.name="квадрат";}
```

Результат работы программы:

Используется объект Окружность

Центр в точке (0, 0), радиус 3

Новая окружность с центром в точке (0, 0) и радиусом 100

