Основы программирования

Основы ООП

Члены класса
Методы

Методы класса

Метод – это член класса, реализующий вычисление или действие, которое может быть выполнено экземпляром или классом.

(Спецификация языка C#)

Методы класса

- Описывают поведение экземпляров класса, то есть те действия, которые может совершать объект или действия, совершаемые над объектом.
- Существуют методы, описывающие поведение класса, а не экземпляра.
- Метод немного похож подпрограмму, только неразрывно связан с тем классом, в котором описан

Синтаксис метода класса

```
атрибуты модификаторы тип_возврата
        Имя_метода (список параметров)
{
  тело_метода;
}
```

Модификаторы метода класса

- `new` – новый метод, переопределяющий метод, унаследованный от предка;
- `static` – статический метод, выполняется в контексте класса, а не объекта;
- `virtual` – виртуальный метод, возможно переопределение в потомках;
- `abstract` – абстрактный метод без реализации, предназначен для задания реализации в потомках;
- `override` – метод в потомке, переопределяющий виртуальный или абстрактный метод, унаследованный от предка;
- `sealed` – метод, переопределение которого в потомках запрещено.

Модификаторы доступа метода класса

- Поскольку метод относится к внешнему представлению и должен быть доступен извне, то чаще всего имеет спецификатор доступа, шире, чем `private`.
- Однако допустимо описывать и приватные методы для использования внутри описываемого класса.

Тип метода класса

- *Тип_возврата* в объявлении метода указывает тип значения, вычисляемого и возвращаемого методом. Для возвращения результата используется оператор `return` с параметром.
- Если метод не возвращает значение, *типом* является `void`.

  Тогда для выхода из метода используется простая форма оператора `return`.
- Если же оператор `return` является последним оператором в теле метода, то его можно опустить.

Имя метода класса

- Это просто *идентификатор.*
- Должно отражать суть действий, выполняемых методом.
- Составляется по нотации Паскаля (с заглавной (=большой) буквы).

Параметры метода класса

- Набор параметров метода с указанием их типов.
- Параметр метода является локальной для метода переменной и действует во всем теле метода наравне с локальными переменными, объявленными в самом теле метода.
- Параметры в метод всегда передаются по значению.

Тело метода

- Это блок операторов, выполняющих действия над объектами.
- Тело метода может быть пустым.
- Методы с пустыми телами часто используются в классах-заглушках.

Сигнатура метода

- Имя метода и список параметров представляют *сигнатуру* метода.
- В классе **не должно быть методов с одинаковыми сигнатурами**, но в классе **может быть группа методов с одинаковыми именами**.
- Разницу в сигнатуры при этом вносит список параметров. Такие методы называются перегруженным.

“Традиционные” методы предоставления доступа: `Get` и `Set`

- Чтение и запись значения поля предоставляется опосредованно, через методы доступа.
- Метод, предназначенный для считывания значения поля, начинается с префикса `Get`. Такие методы называют геттерами.
- Метод для установки значения поля начинается с префикса `Set` и называется сеттером.
- После префикса в имени метода следует название поля, доступ к которому обеспечивает метод. Например: `SetXmin` или `GetSum`.

Классический “геттер” и “сеттер”

```
class MyClass {
  int x = 5;                          //закрытое поле

  public int GetX() {                 //метод-геттер
  return x;                           //возврат значения
     }                                  закрытого поля

  public void SetX(int val){//метод-сеттер
  x = val;                            //задание значения
  }                                    непубличного поля

}
```

“Традиционные” методы преобразования: `To…`

Устоявшимися методами являются также и методы преобразования.

Они начинаются с префикса `To...`, далее следует название того, к чему метод преобразует.

Например:

```
ToString();
ToDouble();
ToInt32();
….
```

Статические методы

Вызываются только по имени класса, без создания объекта этого класса.

Говорят, что статический метод используется в контексте класса, а не в контексте объекта/экземпляра.

Доступ к полям в статическом методе

В теле статического метода, не предполагается доступ к полям объекта напрямую.

Статический метод будет вызван не у объекта (следовательно, не увидит его полей), а у класса.

Следовательно, в статическом методе возможен прямой доступ к тем полям, которые также используются в контексте класса - к константам и к статическим полям.

Использование статических методов

Статический метод предполагает выполнение некоторых общих действий, которые не зависят от конкретного объекта.

Однако они часто используются для выполнения действий над объектом - принимают ссылку на конкретный объект в качестве параметра и могут работать и ним как с его внешним представлением, так и с внутренним состоянием (полями), но через методы доступа.

Ключевое слово `this`

Неявная переменная внутри объекта, ссылающаяся на сам объект, носит имя `this`.

Использование `this`

- Когда параметр метода совпадает по имени и типу с полем объекта. Однако поле – это важная часть объекта, она всегда должна быть доступна для обращения к ней. Эта доступность достигается с помощью ссылки `this`.
- Используется для идентификации объекта, который вызывает на исполнение метод.

Пример использование this

```
class MyClass {
  int a;                          //поле

  public void SetA(int a) //параметр
                                  совпадает по
                                  имени с полем
  {
  this.a = a; //поле осталось доступно
  по ссылке this
  }
}
```