**Глава 4 :ООП**

Что значит this в конструкторе ?

Оно может использоваться внутри любого метода для ссылки на текущий объект. То есть this всегда служит ссылкой на объект, для которого был вызван метод

Что называется полем класса ?

По́ле кла́сса или атрибут в [объектно ориентированном программировании](https://dic.academic.ru/dic.nsf/ruwiki/4923) — переменная, связанная с [классом](https://dic.academic.ru/dic.nsf/ruwiki/68970) или [объектом](https://dic.academic.ru/dic.nsf/ruwiki/117521). Все данные объекта хранятся в его полях. Доступ к полям осуществляется по их имени. Обычно [тип данных](https://dic.academic.ru/dic.nsf/ruwiki/4896) каждого поля задаётся в описании класса, членом которого является поля.

1) Класс — это шаблон или образец, по которому будет создан объект. Обычно класс сравнивают с формой для выпечки печенья, а объект — это само печенье. Конструирование объекта на основе некоторого класса называется получением экземпляра этого класса.

2)

* Инкапсуляция (иногда называемая сокрытием информации) — это ключевое понятие для работы с объектами. Формально инкапсуляцией считается обычное объединение данных и операций над ними в одном пакете и сокрытие данных от других объектов.
* Наследование - при использовании наследования вы говорите: этот новый класс похож на тот старый класс. В коде это пишется как extends, после которого указываете имя базового класса. Тем самым вы получаете доступ ко всем полям и методам базового класса.
* **Полиморфизм** – это перегрузка и переопределение
* Абстракция данных абстра́кция в объектно-ориентированном программировании — это использование только тех характеристик объекта, которые с достаточной точностью представляют его в данной системе.

3) В ООП определены следующие ключевые свойства объектов.

• Поведение объекта — что с ним можно делать и какие методы к нему можно применять.

• Состояние объекта — как этот объект реагирует на применение методов. • Идентичность объекта — чем данный объект отличается от других, характеризующихся таким же поведением и состоянием.

5) Между классами существуют три общих вида отношений.

• Зависимость ("использует — что-то").

• Агрегирование ("содержит — что-то").

• Наследование ("является — чем-то").

6) У конструкторов имеется существенное отличие от других методов: конструктор можно вызывать только в сочетании с операцией new. Конструктор нельзя применить к существующему объекту, чтобы изменить информацию в его полях.

А до тех пор запомните следующее:

• Имя конструктора совпадает с именем класса.

• Класс может иметь несколько конструкторов.

• Конструктор может иметь один или несколько параметров или же вообще их не иметь.

• Конструктор не возвращает никакого значения.

• Конструктор всегда вызывается совместно с операцией new.  
7) Важнее другое: до тех пор, пока метод является закрытым (private), разработчики класса могут быть уверены в том, что он никогда не будет использован в операциях, выполняемых за пределами класса, а следовательно, они могут просто удалить его. Если же метод является открытым (public), его нельзя просто так опустить, поскольку от него может зависеть другой код.

8) Модификатор final удобно применять при объявлении полей простых типов или полей, типы которых задаются неизменяемыми классами. Неизменяемым называется такой класс, методы которого не позволяют изменить состояние объекта. Например, неизменяемым является класс String.

9) Поле с модификатором доступа static существует в одном экземпляре для всего класса. Но если поле не статическое, то каждый объект содержит его копию.

10) Поле является переменной любого типа, которая объявлена непосредственно в [классе](https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/class) или [структуре](https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/struct). Поля являются членами содержащих их типов.

11) Статические методы следует применять в двух случаях.

• Когда методу не требуется доступ к данным о состоянии объекта, поскольку все необходимые параметры задаются явно (например, в методе Math. pow ()).

• Когда методу требуется доступ лишь к статическим полям класса (например, при вызове метода Employee. getNextld ()).  
12) Метод main () не оперирует никакими объектами. На самом деле при запуске программы еще нет никаких объектов. Статический метод main () выполняется и конструирует объекты, необходимые программе.

13) Оба способа конструирования объектов носят общее название перегрузки. О перегрузке говорят в том случае, если у нескольких методов (в данном случае нескольких конструкторов) имеются одинаковые имена, но разные параметры. Компилятор должен сам решить, какой метод вызвать, сравнивая типы параметров, определяемых при объявлении методов, с типами значений, указанных при вызове методов. Если ни один из методов не соответствует вызову или же если одному вызову одновременно соответствует несколько вариантов, возникает ошибка компиляции.

14) Ранее мы рассмотрели два способа инициализации поля:

• установка его значения в конструкторе;

• присваивание значения при объявлении.

15) Поэтому рассмотрим подробнее те действия, которые происходят при вызове конструктора.

1. Все поля инициализируются значениями, предусмотренными по умолчанию (О, fals e или null).

2. Инициализаторы всех полей и блоки инициализации выполняются в порядке их следования в объявлении класса.

3. Если в первой строке кода одного конструктора вызывается другой конструктор, то выполняется вызываемый конструктор.

4. Выполняется тело конструктора.

16) Оператор**new** создает экземпляр указанного класса и возвращает ссылку на вновь созданный объект. :  специальный блок инструкций, вызываемый при создании объекта