



Принципы программирования

YAGNI – YOU AIN’T GONNA NEED IT	Суть принципа: Реализуйте только поставленные задачи! Не делайте то, что не нужно.
KISS – KEEP IT SIMPLE, STUPID	Суть принципа: Люди (включая пользователей продуктов и услуг), как правило, хотят простых вещей, то есть простых в освоении и использовании. Избегайте чрезмерного проектирования и усложнения!
SOLID – SINGLE RESPONSIBILITY PRINCIPLE	Single responsibility principle/Open/closed principle/Liskov substitution principle/Interface segregation principle/Dependency inversion principle.
DRY – Don’t Repeat Yourself	Не повторяйся. Избегай однотипных действий.

Fullstack разработчик на Python

Введение в ООП

Концепция ООП и значимость принципов

Класс	Описание объекта. Другими словами – чертёж по которому мы создаём ту или иную сущность с определённым поведением и качествами.
Объект	это экземпляр с собственным состоянием этих свойств (то, что будет отличать одного человека от другого), любой предмет, существо, явление. Всё, что называется именем существительным, о чём можно сказать «это что-то» или «это кто-то».
Абстракция	Абстракция означает выделение главных, наиболее значимых характеристик предмета и, наоборот, отбрасывание второстепенных, незначительных.
Инкапсуляция	Это свойство системы, позволяющее объединить данные и методы, работающие с ними, в классе и скрыть детали реализации от пользователя.

Модуль Введение в ООП

Наследование	Этот принцип базируется на том, что новый класс описывается на основе уже существующего (родительского), то есть не только перенимает все свойства родительского класса, но ещё и получает новые.
---------------------	---

Классы

Пример создания класса	<pre>class A: pass</pre>
Пример написания метода класса	<pre>def method(self, value): self.value = value</pre>
Пример наследования	<pre>class A(B) # класса A наследуется от класса B.</pre>
Пример множественного наследования	<pre>class A(B, C, ... Z): pass # класс наследуется от сколько угодно других классов</pre>
Создание объекта класса	<pre>O = MyClass(<аргументы конструктора>)</pre>



SKILLFACTORY

Fullstack разработчик на Python

Введение в ООП

Модуль Введение в ООП

Обращение к полям и методам	<code>obj.field</code> # обращение к полю <code>obj.method(<аргументы>)</code> # вызов метода
Геттеры	Не принимают аргументов. Служит для возвращения значения поля. Не принимает аргументов кроме <code>self</code> .
Сеттеры	Устанавливает новое значение в поле. Принимает один аргумент – новое значение. Не должен ничего возвращать.

Магические методы

<code>__add__(self, value)</code>	Переопределяет операцию +
<code>__sub__(self, other)</code>	Переопределяет операцию -
<code>__mul__(self, other)</code>	Переопределяет операцию *
<code>__eq__(self, other)</code>	Переопределяет логическую операцию ==
Полный список методов с примерами	https://docs.python.org/3/reference/datamodel.html