

python

Объектно-ориентированное программирование

Два основных аспекта объектно-ориентированного программирования – классы и объекты. Класс создаёт новый тип, а объекты являются экземплярами класса. Аналогично, когда мы говорим о «переменных типа `int` », это означает, что переменные, которые хранят целочисленные значения, являются экземплярами (объектами) класса `int` .

self

Методы класса имеют одно отличие от обычных функций: они должны иметь дополнительно имя, добавляемое к началу списка параметров. Однако, при вызове метода никакого значения этому параметру присваивать не нужно – его укажет Python. Эта переменная указывает на сам объект экземпляра класса, и по традиции она называется `self`.

Классы

Класс представляет некоторую сущность. Конкретным воплощением класса является объект

Методы объектов

это функции, принадлежащие объекту.

Метод `__init__`

запускается, как только объект класса реализуется. Этот метод полезен для осуществления разного рода инициализации, необходимой для данного объекта.

Переменные класса и объекта

Переменные класса разделяемы – доступ к ним могут получать все экземпляры этого класса. Переменная класса существует только одна, поэтому когда любой из объектов изменяет переменную класса, это изменение отразится и во всех остальных экземплярах того же класса.

Переменные объекта принадлежат каждому отдельному экземпляру класса. В этом случае у каждого объекта есть своя собственная копия поля, т.е. не разделяемая и никоим образом не связанная с другими такими же полями в других экземплярах.

Наследование

Одно из главных достоинств объектно-ориентированного программирования заключается в многократном использовании одного и того же кода, и один из способов этого достичь – при помощи механизма наследования. Легче всего представить себе наследование в виде отношения между классами как тип и подтип.

Метаклассы

Метаклассы существуют для изменения или добавления нового поведения в классы.