

Проектирование с использованием классов

До сих пор в текущей части книги внимание было сосредоточено на применении инструмента ООП на языке Python — класса. Но ООП также связано с *вопросами проектирования*, т.е. с тем, каким образом использовать классы для моделирования полезных объектов. В этой главе мы затронем несколько основных идей ООП и представим ряд дополнительных примеров, более реалистичных, чем многие показанные ранее.

Попутно мы реализуем некоторые распространенные паттерны проектирования в ООП на Python, такие как наследование, композиция, делегирование и фабрики. Мы также исследуем концепции классов, ориентированные на проектирование, вроде псевдозакрытых атрибутов, множественного наследования и связанных методов.

Одно заблаговременное примечание: некоторые упоминаемые здесь термины, относящиеся к проектированию, требуют большего объема пояснений, чем я смог предложить в книге. Если данный материал вызвал у вас интерес, тогда я рекомендую в качестве следующего шага почитать какую-нибудь книгу по объектно-ориентированному проектированию или паттернам проектирования. Как вы вскоре убедитесь, хорошая новость заключается в том, что Python делает многие традиционные паттерны проектирования тривиальными.

Python и объектно-ориентированное программирование

Давайте начнем с обзора — реализация ООП в языке Python может быть сведена к трем идеям.

Наследование

Наследование основано на поиске атрибутов в Python (в выражениях `X.name`).

Полиморфизм

В выражении `X.method` смысл `method` зависит от типа (класса) подчиненного объекта `X`.

Инкапсуляция

Методы и операции реализуют поведение, хотя сокрытие данных по умолчанию является соглашением.