

В текущем разделе будет представлена общая *архитектура* программы Python — способ разделения программы на коллекцию файлов исходного кода (также известных как модули) и связывания частей в единое целое. Вы увидите, что Python стимулирует модульную структуру программ, которая группирует функциональность в связанные и многократно применяемые единицы естественными и почти автоматическими способами. Одновременно мы также исследуем основные концепции модулей Python, импортирования и атрибутов объектов.

Структурирование программы

На самом базовом уровне программа Python состоит из текстовых файлов, содержащих *операторы* Python, с одним главным файлом *верхнего уровня* и нулем или большим количеством добавочных файлов, известных как *модули*.

Посмотрим, как все работает. Файл верхнего уровня (он же сценарий) содержит главный поток управления программы — именно данный файл запускается для старта приложения. Файлы модулей являются библиотеками инструментов, используемых для сбора компонентов, которые применяются файлом верхнего уровня и возможно где-нибудь еще. Файлы верхнего уровня используют инструменты, определенные в файлах модулей, а модули применяют инструменты, определенные в других модулях.

Несмотря на то что файлы модулей тоже относятся к файлам кода, они обычно ничего не делают, когда запускаются напрямую; взамен в них определяются инструменты, предназначенные для использования в других файлах. Файл *импортирует* модуль для получения доступа к определенным в нем инструментам, которые известны как его *атрибуты* — имена переменных, присоединяемые к таким объектам, как функции. В конечном счете, для работы с инструментами мы импортируем модули и обращаемся к их атрибутам.

Импортирование и атрибуты

Давайте перейдем к чуть большей конкретике. На рис. 22.1 показана упрощенная структура программы Python, состоящей из трех файлов: `a.py`, `b.py` и `c.py`. Файл `a.py` выбран в качестве файла верхнего уровня; он будет простым текстовым файлом с операторами, которые при его запуске выполняются от начала до конца. Файлы `b.py` и `c.py` — это модули; они являются простыми текстовыми файлами, также содержащими операторы, но их операторы, как правило, не выполняются напрямую. Как объяснялось ранее, модули обычно импортируются другими файлами, которые заинтересованы в применении инструментов, определенных в модулях.

Предположим, что в файле `b.py` на рис. 22.1 определена функция по имени `spam`, предназначенная для внешнего применения. Согласно тому, что уже известно в результате исследования функций в части IV, файл `b.py` будет содержать оператор `def` языка Python для создания объекта функции, который позже можно запускать, передавая ему ноль или большее количество значений в круглых скобках после имени функции:

```
def spam(text):          # Файл b.py
    print(text, 'spam')
```

Теперь допустим, что в `a.py` желательно использовать `spam`. Для этой цели файл может содержать операторы Python следующего вида:

```
import b                 # Файл a.py
b.spam('gumby')          # Выводит gumby spam
```