

базы данных с постоянным хранением. С применением модулей `pickle` или `shelve` из Python экземпляры классов можно сохранять на диске за один шаг. Мы использовали `shelve` для хранения экземпляров классов в учебном руководстве по ООП в главе 28, но интерфейс `pickle` также удивительно легко применять с объектами:

```
import pickle
object = SomeClass()
file = open(filename, 'wb')    # Создать внешний файл
pickle.dump(object, file)      # Сохранить объект в файле

import pickle
file = open(filename, 'rb')
object = pickle.load(file)     # Извлечь объект в более позднее время
```

Модуль `pickle` преобразует объекты из памяти в сериализованные потоки байтов (строки в Python), которые можно хранить в файлах, отправлять по сети и т.д.; распаковка с помощью `pickle` преобразует потоки байтов в идентичные объекты в памяти. Модуль `shelve` похож, но сохраняет объекты, автоматически обработанные `pickle`, в базе данных с доступом по ключу, которая экспортирует словарный интерфейс:

```
import shelve
object = SomeClass()
dbase = shelve.open(filename)
dbase['key'] = object          # Сохранить под ключом

import shelve
dbase = shelve.open(filename)
object = dbase['key']         # Извлечь объект в более позднее время
```

В нашем примере с пиццерией использование классов для моделирования сотрудников означает возможность создания простой базы данных за счет выполнения небольшой дополнительной работы. Сохранение таких объектов экземпляров в файле с помощью `pickle` делает их постоянными между запусками программы на Python:

```
>>> from pizzashop import PizzaShop
>>> shop = PizzaShop()
>>> shop.server, shop.chef
(<Employee: name=Pat, salary=40000>, <Employee: name=Bob, salary=50000>)
>>> import pickle
>>> pickle.dump(shop, open('shopfile.pkl', 'wb'))
```

Здесь в файле сохраняется весь составной объект `shop` целиком. Чтобы позже поместить его в другой сеанс или программу, также достаточно одного шага. В действительности объекты, восстановленные подобным образом, предохраняют свое состояние и поведение:

```
>>> import pickle
>>> obj = pickle.load(open('shopfile.pkl', 'rb'))
>>> obj.server, obj.chef
(<Employee: name=Pat, salary=40000>, <Employee: name=Bob, salary=50000>)
>>> obj.order('LSP')
LSP orders from <Employee: name=Pat, salary=40000>
Bob makes pizza
oven bakes
LSP pays for item to <Employee: name=Pat, salary=40000>
```

Это просто запускает эмуляцию в существующем виде, но мы могли бы расширить модель пиццерии, чтобы отслеживать запасы, выручку и т.п. — ее сохранение в файле после внесения изменений предохранит обновленное состояние. Дополнительные сведения о модулях `pickle` и `shelve` ищите в руководстве по стандартной библиотеке, а также в главах 9 (первого тома), 28 и 37.