Перед нами стоит следующая задача: Составить программу, которая создаёт резервные

копии всех наших важных файлов.

Хотя задача и проста, информации явно недостаточно, чтобы приступать к её решению.

Необходим некоторый дополнительный анализ.

Например,

как мы выберем, какие файлы необходимо копировать? Как их хранить? Где их хранить?

После надлежащего анализа мы проектируем нашу программу. Мы создаём список, опи-

сывающий то, как наша программа должна работать. В данном случае я создал список

того, как я себе представляю её работу. Когда вы проектируете программу, у вас может по-

лучиться другой результат, поскольку каждый человек представляет себе это по-своему,

так что это в порядке вещей.

РАЗБИЛИ НА ПОДЗАДАЧИ:

1. Файлы и каталоги, которые необходимо скопировать, собираются в список.

2. Резервные копии должны храниться в основном каталоге резерва.

3. Файлы помещаются в zip-архив.

4. Именем для zip-архива служит текущая дата и время.

5. Будем использовать стандартную команду zip, имеющуюся по умолчанию в любом

стандартном дистрибутиве GNU/Linux. Пользователи Windows могут установить её

со страницы проекта GnuWin32 и добавить «C:\Program Files\GnuWin32\bin» к

системной переменной окружения PATH, аналогично тому, как мы это делали для

самой команды «python». Обратите внимание, что для этого подойдёт любая коман-

да архивации, если у неё есть интерфейс командной строки, чтобы ей можно было

передавать аргументы из нашего сценария.

В процессе создания программы мы прошли через несколько стадий. Эти стадии можно

свести примерно в такой список:

• Что (Анализ)

• Как (Проектирование)

• Создание (Реализация)

• Тестирование (Тестирование и Отладка)

• Использование (Развёртывание и Оперирование)

• Поддержка (Усовершенствование)

Процедура, которую мы прошли при написании сценария создания резервных копий ре-

комендуется и для других программ: Проведите анализ и проектирование. Начните ре-

ализацию с простейшей версии. Протестируйте и отладьте её. Попользуйтесь ею, чтобы

убедиться, что она работает, как ожидалось. После этого добавляйте любые необходимые

функции, повторяя цикл «Создание-Тестирование-Использование» столько раз, сколько

потребуется. Помните, Программы выращиваются, а не строятся.

C