

Домашнее задание 3. Время упаковки.

Срок сдачи: 16 марта, 2015

Прежде чем приступить к написанию, прочтите раздел **Замечания**.

1 Условие

На вход программе в качестве первого аргумента дается код операции архивации/распаковки/печати ("compress" / "decompress" / "list").

При архивации, второй аргумент – имя выходного файла. Остальные аргументы соответствуют относительным путям к файлам и/или директориям, которые подлежат упаковке. Также в списке аргументов могут встретиться адреса web-страниц – они обязательно начинаются с http:// (например, http://yandex.ru)

Все файлы, лежащие в поддеревьях файловой системы, корнями которых являются указанные пути, а также содержимое web-страниц должно быть объединено в один файл (архив), **сжатый алгоритмом zip**. Архив необходимо организовать приблизительно так же как “сообщения” из первого дз:

- Относительный путь до первого файла
- Кол-во байт первого файла
- Содержимое первого файла
- ...

Т.е. все данные **сначала** объединяются по указанной в задании схеме, а потом полученная информация **целиком** архивируется алгоритмом zip.

При распаковке, вторым аргументом является имя файла с архивом. В папке, из которой произведен запуск, должна быть воссоздана исходная структура каталогов (ну и про сами файлы, естественно не забудьте). Web-страницы при распаковке должны оказаться в отдельной папке http, имена файлов соответствуют адресам web-страниц, в которых символ «/» заменен на «_»

При печати, вторым аргументом является имя файла с архивом. Программа должна распечатать структуру дерева каталогов аналогичную той, которая была бы создана при распаковке, за исключением папок и файлов, название которых начинается с символа «.» (в Unix это “скрытые” файлы).

Дерево должно распечатываться в консоль в следующем формате:

```
|_subfolder1
|           |_subfolder 1.1
|           |           |_a.txt
|           |_subfolder 1.2
|_subfolder2
|           |_a.txt
|           |_b.txt
```

2 Замечания

- Для архивации посмотрите на классы из стандартной библиотеки. Например, ZipInputStream/ZipOutputStream
- Для доступа к содержимому web-страниц можете воспользоваться классом URLConnection. Обратите внимание на метод getInputStream().
- Нельзя использовать системные вызовы

- Нельзя использовать дополнительные библиотеки, кроме стандартной
- Не забудьте о выводе осмысленной информации на консоль (в том числе об отсутствии доступа к той или иной директории/файлу)
- Файлы, к которым нет доступа (**File.canRead()** вернул **false** или произошел **SecurityException**) или web-страницы, к которым не удастся получить доступ, необходимо пропускать (при этом выводить информацию об этом), но не прерывать работу программы.
- В случае, если в процессе работы произошел IOException разрешается прерывать работу программы, но дополнительные баллы будут даны, если соответствующий файл (или веб-страницы) будет пропущен (и выведена соответствующая информация пользователю) при сохранении валидности архива.
- Директории, в поддеревьях которых нет ни одного файла, объявляются бесполезными и нещадно игнорируются =)
- Нелишним будет почитать немного про java security policy (в частности **SecurityManager** и **Security Exception**).
- Ваш код будет тестироваться нами на Linux-е и Windows-е, папках с различными правами доступа и (вы/в)ключенной опции **-Djava.security.manager**
- Тестируйте тщательно!