

Input and Output

IMPORTANT! Completed tasks are evaluate at the final result and the manner of applying the knowledge gained in the course

Required reading

- <http://docs.oracle.com/javase/tutorial/essential/io/index.html>
- <https://github.com/square/okio>
- <https://developers.google.com/protocol-buffers/>

Remark: Конзолния изход трябва да се използва единствено в main'a.

Задачи

1. Да се направи клас, който чете от конзолата. Трябва да има следните методи:
 - `readString();`
 - `readInt();`
 - `readChar();`
 - `readFloat();`
2. Да се направи приложение, който чете текст от конзолата и записва въведеното в текстов файл.
 - Четенето спира при въвеждане на ред съдържащ само една точка ".". В началото да се указва името на създавания файл.
3. Да се направи приложение, който реверсира съдържанието на текстов файл.
 - Съдържанието на текстовия файл се прочита и се слага в буфер. Съдържанието на буфера се реверсира и след това се записва като ново съдържание на файла.
Text File Content:

```
1234567890  
abcdefghij
```

Output 1:

```
0987654321  
jihgfedcba
```

Output 2:

```
jihgfedcba  
0987654321
```

И 2'та резултата са валидно решение на задачата.

4. Да се направи клас `TransferObject`, който служи за прехвърляне на съдържанието от един поток в друг.

- Да се се позволява частично прехвърляне

Опътване: `TransferObject#transfer(InputStream in, OutputStream out, int offset, int size)`

Example for partial transfer:

```
Input: 1234567890  
Transfer 4 bytes from offset 2  
Output: 3456
```

5. Да се направи клас `DataClass`, съдържащ методите `saveObject` и `getObject`, и друг произволен клас, който да може да се сериализира. Пояснение за методите на `DataClass`:

```
public class DataClass {  
    /**  
     * Saves the provided object "o" in the output stream.  
     */  
    public void saveObject(OutputStream out, Object o) throws IOException {  
        ...  
    }  
  
    /**  
     * Reads an object from the provided InputStreams and returns it as result.  
     */  
    public Object getObject(InputStream in) throws IOException {  
        ...  
    }  
}
```

Опътване: Да се прочете за класа `Serializable` и да се определят предимствата и недостатъците му.

6. Да се направи клас `DirectoryBrowser`, който да извежда съдържанието на директория от файловата система.

- Класът има метод `listContent(String path)`.
- Параметърът `path` е името на директория или файл с пълния път преди него (пример: `H:\development\`, `C:\MyFile.jpg`)
- Методът проверява дали `path` сочи към файл или директория
- Ако сочи към файл - извежда се съобщение, че е подадено име на файл. Ако сочи към директория - извежда се списък с имената на всички файлове и директории, съдържащи се в нея
- Пример: `path= H:\development\`, съдържа директориите `Proj1`, `MyProj` и файлът `Proba.txt`