

CISC124 – Today’s Topics

- Assignment 2
- Text files
- Text files input/output
- Binary files input/output

Winter 2019

F. de la Parra

1

•1

Assignment2 Comments

- “Item” references have been changed to “Term” references in description document
- Typo in query lines has been fixed (file queries.txt). For example, a query line in file “queries.txt” is now: query: HP:1234567
- Terms having an is_obsolete keyword should be skipped (216 items)
- When reading the “Terms” in, the keyword values that absolutely need to be parsed an isolated are: id and is_a
- The entire content of a term can be captured in a single String attribute of an object of type Term
- For “Terms” having two or more is_a keywords (3163 Terms), select just the first instance to identify a parent “Term”

Winter 2019

F. de la Parra

2

•2

Text files

- Sequence of lines terminated by EOF (End of File).
- Each line is string of readable characters terminated by CR (Carriage Return) LF (Line Feed)
- Access is sequential → reading or writing a line advances the “line pointer” by 1. Cannot go back to previous lines
- Can open file for “read” or “write”, and usual processing steps are:
 - Open in “read” or “write” mode. Pointer is set to first line of the file
 - Process lines, one by one, from the first line until the EOF
 - Close the file

WRITE

```
while (Process not finished) {
    Write Lines
    Is process finished?
}
```

READ

```
while (Not at EOF) {
    Read Line(s)
    Process Lines
}
```

Winter 2019

F. de la Parra

3

•3

Opening and using files

- Constructor methods to open a file throw exceptions.
- Operations to open a file must be in a try-catch block
- FileNotFoundException is the usual one

- Reading a line from a file might throw exception
- NoSuchElementException when using a Scanner object is a typical one

Winter 2019

F. de la Parra

4

•4

Classes (write to text file)

Classes for file Input/Output are java.io package

- Write to a text file
 - java.io.FileOutputStream
 - java.io.PrintWriter

Strings

- Object of type FileOutputStream implements a data stream to a file
- Constructor may throw a FileNotFoundException
- Can open a file in overwrite or append mode:
 - `public FileOutputStream(String name, boolean append)`

- Object of type PrintWriter supports high level methods to manipulate text data: print, println, printf, write

Bytes

Winter 2019

F. de la Parra

5

•5

Classes (read from text file)

- Read from a text file
 - java.io.FileInputStream
 - java.util.Scanner

Strings

- Object of type FileInputStream implements a data stream from a file
- Constructor may throw a FileNotFoundException
- Can open a file pointed to by a FilePath or an object of type File:
 - `public FileInputStream(String FilePath)`

- Object of type Scanner supports high level methods to parse text data (even using delimiters): next, nextInt, hasNextInt, hasNext, nextLine

Bytes

Winter 2019

F. de la Parra

6

•6

Binary files

- Opening, reading/writing and closing schemes similar to what has been explained for text files
- Files usually used to save serialized objects and read back into a program deserialized objects
- The difference with text files is that data streams to read/write to binary files are byte-oriented
- IOException must be caught not only when opening these files, but also when reading/writing to them
- Data read in can also cause exceptions (e.g., when reading objects → ClassNotFoundException)
- Classes involved:
 - Write to file → ObjectOutputStream
 - Read from file → ObjectInputStream

Winter 2019

F. de la Parra

7

•7