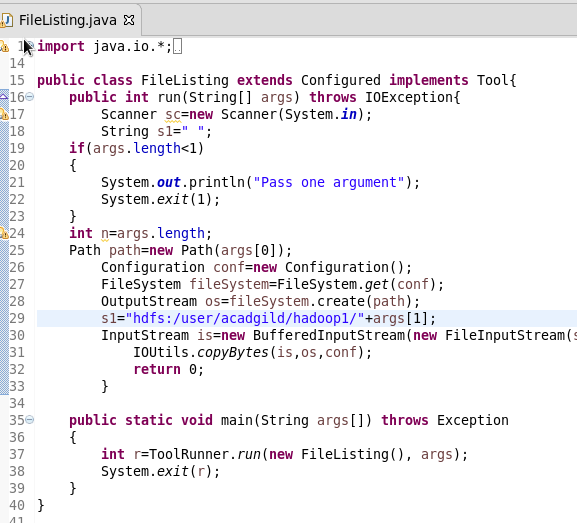
Problem 4.4)



Problem 2)

a)

DFSInputStream:

 Handles communication of the namenode with various datanodes. • Handles integrity of data contained by the blocks. • Manages data read activity in case of datanode failure. • Called internally by FSDataInputStream.

b)

DFSOutputStream:

Handles communication of the namenode with various datanodes.

Called internally by FSDataOutputStream.

c)

FSDataInputStream:

A stream that wraps another stream it means that any operation done on a wrapper stream is in turn transformed into operations into the wrapped streams.

The wrapped stream can be a file, a memory region, a network connection. FSDataInputStream wraps the DataInputStream and implements Seekable, PositionedReadable interfaces which provide method like getPos(), seek() method to provide Random Access on HDFS file.

d)

FSDataOutputStream:

Filesystem’s create () method return FSDataOutputStream, which use to create new HDFS file or write the content at the EOF. It doesn’t provide seek because of HDFS limitation to write to content at the EOF only. It wrap Java IO’s DataOutputStream and add method such as getPos() to get the position of the file and write() to write the content at the last position.Below method signature provide FSDataOutputStream:  
Create method on FileSystem create file e.g.

public FSDataOutputStream create(Path f) create empty file.

public FSDataOutputStream append(Path f) will append existing file

Create method also pass Progressable interface to track the status during file creation.

public FSDataOutputStream create(Path f, Progressable progress)