```
public class Indexer {

  public static void main(String[] args) throws Exception {
    if (args.length != 2) {
      throw new IllegalArgumentException("Usage: java " +
➥Indexer.class.getName()
        + " <index dir> <data dir>");
    }
    String indexDir = args[0];
    String dataDir = args[1];

    long start = System.currentTimeMillis();
    Indexer indexer = new Indexer(indexDir);
    int numIndexed;
    try {
      numIndexed = indexer.index(dataDir, new TextFilesFilter());
    } finally {
      indexer.close();
    }
    long end = System.currentTimeMillis();

    System.out.println("Indexing " + numIndexed + " files took "
      + (end - start) + " milliseconds");
  }

  private IndexWriter writer;

  public Indexer(String indexDir) throws IOException {
    Directory dir = FSDirectory.open(new File(indexDir));
    writer = new IndexWriter(dir,
                 new StandardAnalyzer(
                     Version.LUCENE_30),
                 true,
                 IndexWriter.MaxFieldLength.UNLIMITED);
  }

  public void close() throws IOException {
    writer.close();
  }

  public int index(String dataDir, FileFilter filter)
    throws Exception {

    File[] files = new File(dataDir).listFiles();

    for (File f: files) {
      if (!f.isDirectory() &&
          !f.isHidden() &&
          f.exists() &&
          f.canRead() &&
          (filter == null || filter.accept(f))) {
        indexFile(f);
      }
    }

    return writer.numDocs();
  }

  private static class TextFilesFilter implements FileFilter {
    public boolean accept(File path) {
      return path.getName().toLowerCase()
             .endsWith(".txt");
    }
  }
```

1 Create index in this directory

2 Index *.txt files from this directory

3 Create Lucene IndexWriter

4 Close IndexWriter

5 Return number of documents indexed

6 Index .txt files only, using FileFilter

```
  }

  protected Document getDocument(File f) throws Exception {
    Document doc = new Document();
    doc.add(new Field("contents", new FileReader(f)));
    doc.add(new Field("filename", f.getName(),
                      Field.Store.YES, Field.Index.NOT_ANALYZED));
    doc.add(new Field("fullpath", f.getCanonicalPath(),
                      Field.Store.YES, Field.Index.NOT_ANALYZED));
    return doc;
  }

  private void indexFile(File f) throws Exception {
    System.out.println("Indexing " + f.getCanonicalPath());
    Document doc = getDocument(f);
    writer.addDocument(doc);
  }
}
```

7 Index file content

8 Index filename

9 Index file full path

10 Add document to Lucene index