





[illegible]

### 3. Lucene

```
Lucene00CCCCCCCCCCCCCCCCCCCCCCCCCC(
++)CCCCCCCCCCCCCCCCjavaCCCCCCCCCCCCLuceneCCCCCCCC(CCCCCC
CCCCCCCC)CCCCCCCCCCCC
```

- `BooleanQuery, SpanQuery, TermQuery...`  
`Query`  
`Query`  
`Lucene` (`HitCollector, ...`)  
`AO`  
`P`.
- `hasNext()`, `next()`  
`.`  
`.`

4.

```

    Lucene, Similarity
    , Lucene Lucene
    tf/idf Lucene
    Lucene

```

5.

```

    Lingway, Lucene
    Lucene

```

6.   □ □ □ □ □ □ □ □ □ □

```

    DirectorySolrNutch+Hadoop
SolrNutchLuceneLuceneSolrNutch
..HadoopLucene:Lucene
, LuceneLuce

```

ne 爬取数据并存储在 Solr 中。Nutch 是一个开源的搜索引擎，它支持 Nutch 爬取数据并存储在 Solr 中。

### 3. 第 4 章

#### 3.1 第 4 章

第 4 章

Lucene 是一个开源的搜索引擎，它支持 Lucene 爬取数据并存储在 Solr 中。

Lucene-core lucene-analyzers-common lucene-highlighter lucene-memory lucene-queryparse

第 4 章

第 4 章 lucene 是一个开源的搜索引擎，它支持 Lucene 爬取数据并存储在 Solr 中。

第 4 章

#### 3.2 第 4 章

Lucene 是一个开源的搜索引擎，它支持 Lucene 爬取数据并存储在 Solr 中。

List<Field>, Java 是一个开源的搜索引擎，它支持 Lucene 爬取数据并存储在 Solr 中。

第 4 章

第 4 章

1. 第 4 章 IndexWriter

2. 第 4 章 Document

3. 第 4 章 Document

4. 第 4 章

第 4 章

```
package com.lucene.test;

import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
```

```
import org.apache.lucene.analysis.Analyzer;
import org.apache.lucene.analysis.standard.StandardAnalyzer;
import org.apache.lucene.document.Document;
import org.apache.lucene.document.Field;
import org.apache.lucene.document.FieldType;
import org.apache.lucene.document.TextField;
import org.apache.lucene.index.IndexWriter;
import org.apache.lucene.index.IndexWriterConfig;
import org.apache.lucene.store.Directory;
import org.apache.lucene.store.FSDirectory;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.baidu.entity.Item;
import com.baidu.service.ItemService;

public class CreateIndex {
    public static void main(String[] args) {
        ApplicationContext context = new
ClassPathXmlApplicationContext("spring.xml");
        ItemService service = context.getBean(ItemService.class);
        //□□□□□□□□□□□□□□□□□□□□
        List<Item> list = service.findAllItems();

        try {
            //□□□□□□□□□□□□□□□□□□□□
            Directory directory = FSDirectory.open(new
File("E:/dicindex").toPath());
            //□□□□
            Analyzer analyzer = new StandardAnalyzer();
            IndexWriterConfig config = new IndexWriterConfig(analyzer);
            //1□□□□□□□□□□□□□□□□□□□□
            IndexWriter writer = new IndexWriter(directory,config);

            //□□□□□□□□□□□□□□□□□□□□
            List<Document> dlist = new ArrayList<Document>();

            //□□□□□□□□□□□□□□□□□□□□
            for (Item item : list) {
                /*
                 * TYPE_STORED      □□□□□□□□□□
                 * TYPE_NOT_STORED  □□□□□□□□□□
                 */
                Document document = new Document();
```



7 8

```
//□□□□□□□□□□□□document□□
Document doc = search.doc(index);
//□Document□□□□□□□□□□
Item a = new Item();
a.setId(new BigInteger(doc.get("id")));
a.setTitle(doc.get("title"));
a.setSellPoint(doc.get("content"));
list.add(a);
}
//□□□□□□□
for (Item article : list) {
    System.out.println(article.getTitle());
}
//□□□□
```

This document was truncated here because it was created in the Evaluation Mode.