

目录

FileSearch 项目实战

一、项目介绍.....	1
1.1 项目简介.....	1
1.2 项目功能.....	1
1.3 项目展示.....	1
二、项目搭建和部署.....	2
2.1 项目框架设计.....	2
2.2 项目环境和配置.....	3
2.3 项目技术.....	3
三、项目开发.....	3
3.1 项目开发的流程.....	3
3.2 项目模块介绍.....	4
四、项目测试.....	6
4.1 测试准备.....	6
4.2 测试执行.....	7
五、项目总结.....	8

FileSearch 项目实战

一、项目介绍

1.1 项目简介

搜索 Everything 软件是一款强大的文件搜索软件，它是基于 NTFS 文件系统的 USN Journal，利用操作记录的文件操作日志来进行搜索，特点是效率高，速度快，但仅用于 NTFS 文件系统。

本项目使用实时的本地文件进行搜索，保存文件信息以便于提高搜索效率。同时在实现基础的文件搜索功能上，又进行了部分优化。

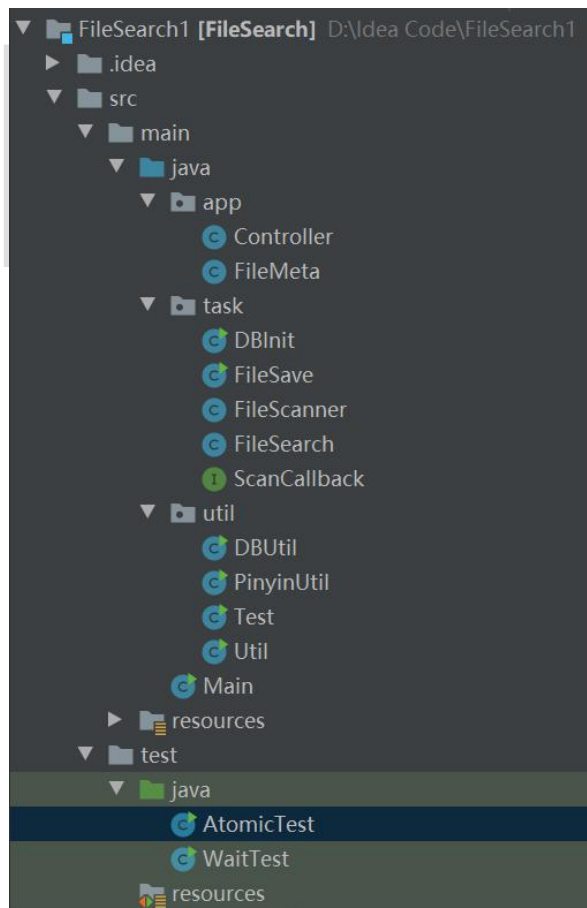
相较于搜索 Everything，本项目的优点在于使用实时的本地文件进行搜索，文件搜索支持汉语拼音的搜索(全拼和首拼)。

1.2 项目功能

- 1、获取：选择文件夹，获取当前文件夹下的所有文件和文件夹。
- 2、搜索：输入文件名称（支持汉语拼音全拼和首拼搜索）。
- 3、更新：当更新文件信息时，数据库实时更新。

1.3 项目展示

选择文件夹/文件搜索

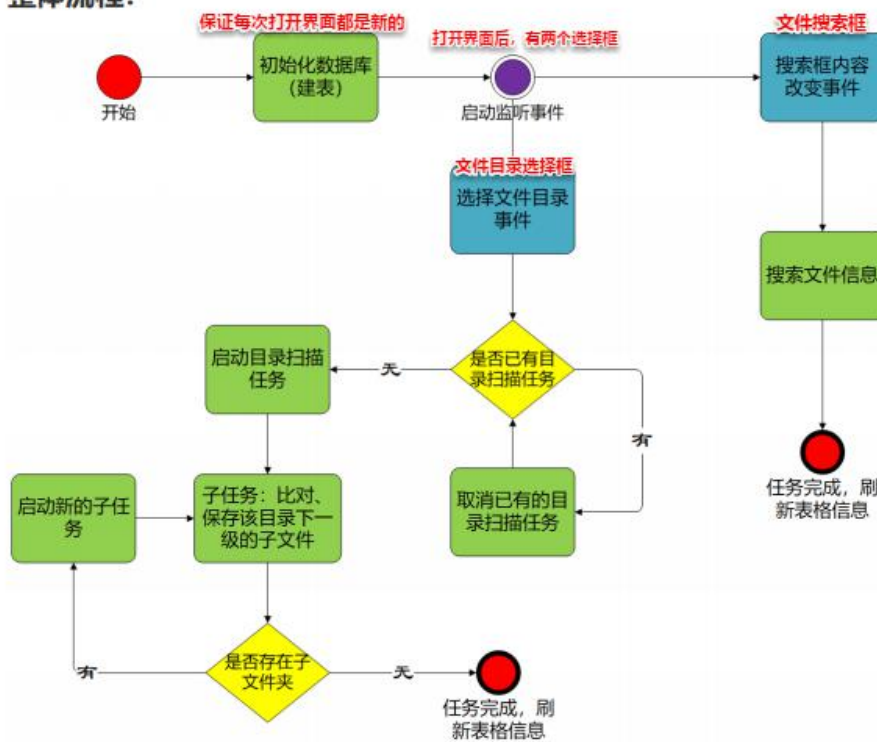


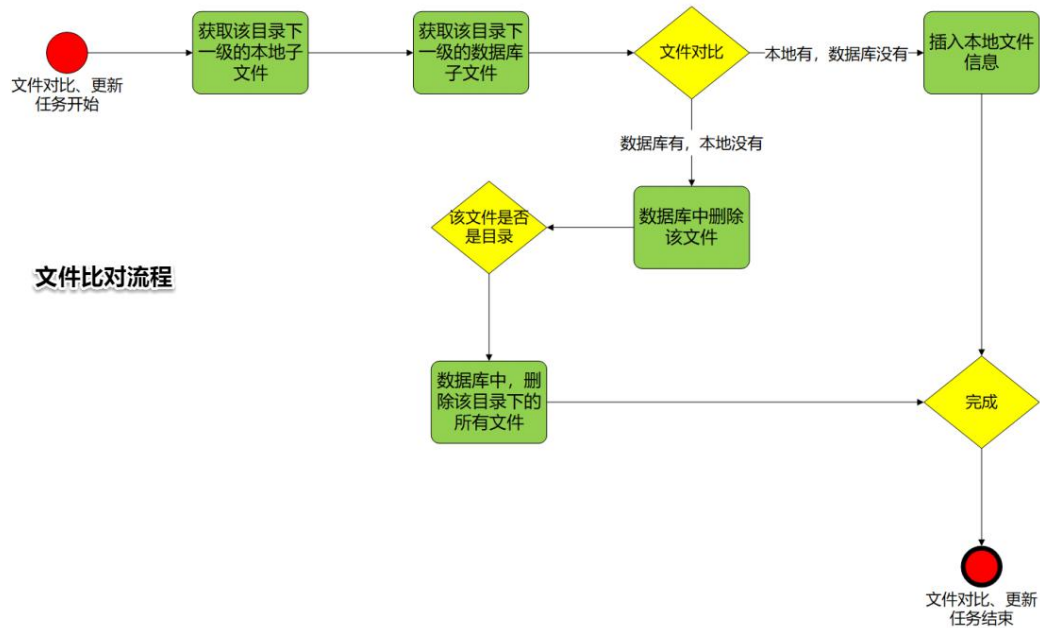


二、项目搭建和部署

2.1 项目框架设计

整体流程:





2.2 项目环境和配置

1、IDEA—Java 项目开发工具

仿 Everything 的基础功能的开发。

2、Maven—项目管理工具

Maven 的配置和使用。

2.3 项目技术

1、多线程

多线程目录扫描，提高文件搜索效率。

2、SQLite 和 JDBC

简介：SQLite 是一款轻量级的嵌入式内存数据库（嵌入在进程中，运行在内存中的数据库）。

特点：可跨平台、轻量级、无需安装、配置及管理、允许多个线程安全访问、支持 Java 等多种开发语言。

JDBC: Java 数据库连接，用来规范客户端程序访问数据库的应用程序接口。

3、pinyin4j

简介：Java 的一个内库，提供对中文汉字到拼音的转换。

利用 pinyin4j 实现文件搜索的全拼搜索和首拼搜索功能。

4、JavaFX

简介：JavaFX 是一个强大的图形和多媒体处理工具包集合。

特点：跨平台、提供对 Web 组件支持（HTML5 的支持、CSS 样式支持等）、支持拖拽式界面开发。

三、项目开发

3.1 项目开发的流程

1、数据库和数据表

DBInit：数据库和数据表初始化

DBUtil：数据库连接和释放

2、监听事件,选择执行功能 Controller

(1) 选择文件目录事件

FileScanner：扫描文件

FileSave：对比更新文件

(2) 搜索框事件

FileMeta：文件属性的设置(文件名、文件路径、文件大小、更新时间)

FileSearch：文件搜索功能

PinyinUtil：首拼和全拼

3.2 项目模块介绍

1、数据库模块

初始化数据库,添加搜索框监听器;

```
public void initialize(URL location, ResourceBundle resources) {

    // 界面初始化时，需要初始化数据库及表

    DBInit.init();

    // 添加搜索框监听器，内容改变时执行监听事件

    searchField.textProperty().addListener(new ChangeListener<String>() {
        public void changed(ObservableValue<? extends String> observable, String
oldValue, String newValue) {
            freshTable();
        }
    });
}
```

连接数据库,执行 Sql 语句,释放数据库

```
public static void init(){

    // 数据库 jdbc 操作：sql 语句的执行

    Connection connection = null;
    Statement statement = null;
    try {

        // 1.建立数据库连接 Connection

        connection = DBUtil.getConnection();

        // 2.创建 sql 语句执行对象 Statement

        statement = connection.createStatement();
        String[] sqls = readSQL();
        for(String sql : sqls){

            System.out.println("执行 sql 操作："+sql); // 3.执行 sql 语句

            statement.executeUpdate(sql);

        }

        // 4.如果是查询操作，获取结果集 ResultSet，处理结果集
```

```

    }catch (Exception e){
        e.printStackTrace();

        throw new RuntimeException("初始化数据库表操作失败", e);
    }finally {

        // 5.释放资源

        DBUtil.close(connection, statement);
    }
}

```

2、搜索模块

输入搜索关键字设置

```

public static String[] get(String name){
    String[] result = new String[2];

    StringBuilder pinyin = new StringBuilder();//全拼 //线程不安全

    StringBuilder pinyinFirst = new StringBuilder();//拼音首字母

    for(char c : name.toCharArray()){//遍历后转字符串数组

        try {
            String[] pinging = PinyinHelper
                .toHanyuPinyinStringArray(c, FORMAT);

            //toHanyuPinyinStringArray 静态方法，将汉语与汉语拼音转化。如果不是汉语参
            //数，会返回 null

            if(pinging == null || pinging.length == 0){//没有解析到任何字符

                //添加原始字符

                pinyin.append(c);
                pinyinFirst.append(c);
            }else{

                //全拼

                pinyin.append(pinging[0]);

                //拼音首字母

                pinyinFirst.append(pinging[0].charAt(0));
            }
        } catch (BadHanyuPinyinOutputFormatCombination
badHanyuPinyinOutputFormatCombination) {
            pinyin.append(c);

```

```

        pinyinFirst.append(c);
    }
}
result[0] = pinyin.toString();
result[1] = pinyinFirst.toString();
return result;
}

```

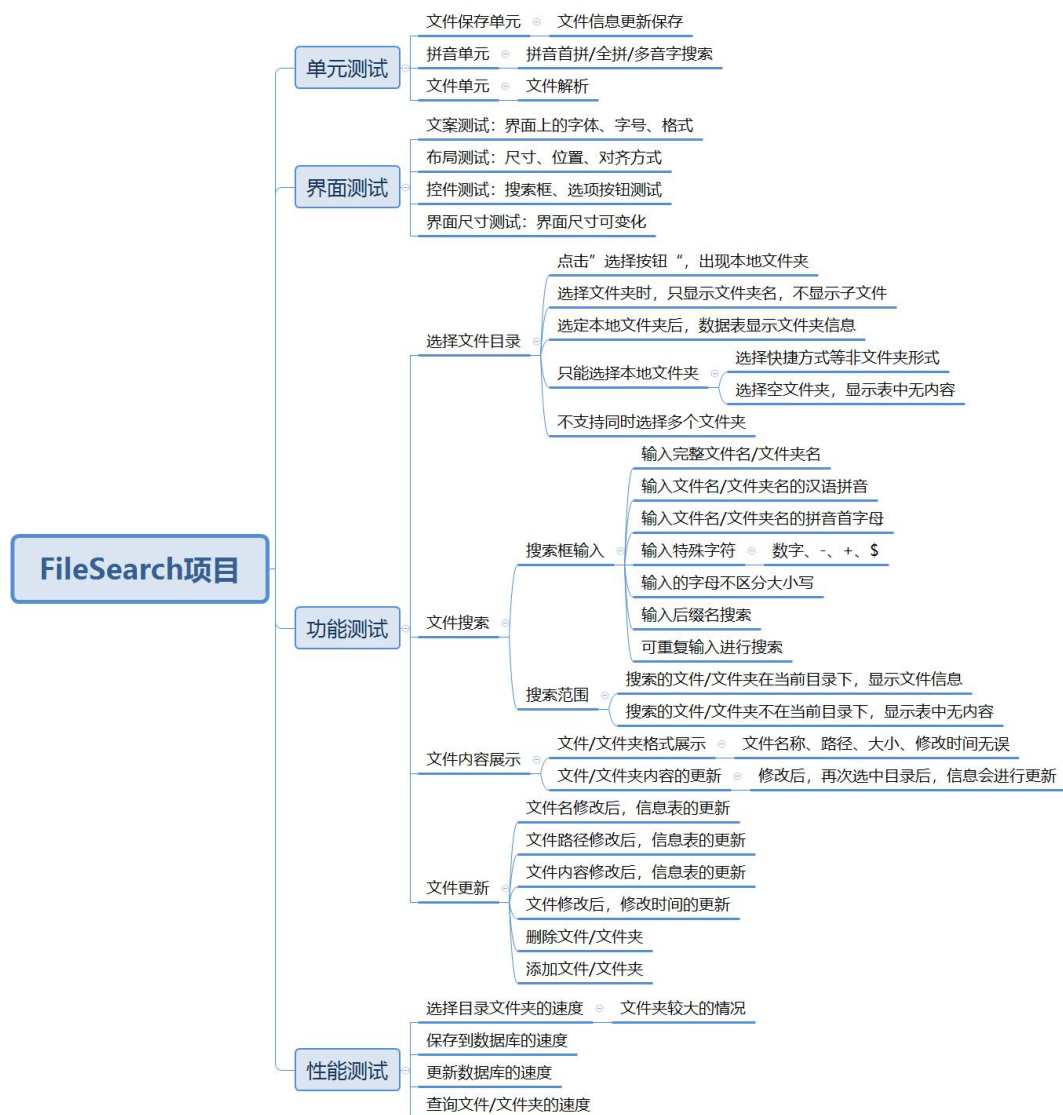
四、项目测试

4.1 测试准备

1、测试需求分析

- (1) 打开界面,数据库会进行初始化,未发现任何文件信息;
- (2) 确定功能事件后,数据库建立连接;
- (3) 选择文件目录功能,数据库进行检索并显示当前目录下的所有文件夹和文件;
- (4) 在搜索框输入搜索信息(进行全拼首拼输入),搜索出文件信息;
- (5) 当有文件或文件夹更新时,再次选择目录会发现信息已更新;

2、测试策略



4.2 测试执行

1、单元测试

(1) DBInit：读取数据库文件代码测试

```
//测试

public static void main(String[] args) {
    String[] sols = readSQL();
    for(String sql : sols){
        System.out.println(sql);
    }

    //测试

    System.out.println(Arrays.toString(readSQL()));
    init();
}

Run: DBInit x
"C:\Program Files\Java\jdk1.8.0_261\bin\java.exe" ...
create table if not exists file_meta(    name varchar(50) not null,    path va
[create table if not exists file_meta(    name varchar(50) not null,    path v
获取数据库文件路径 : jdbc:sqlite://D:\Idea Code\FileSearch1\target\FileSearch.db
执行sql操作 : create table if not exists file_meta(    name varchar(50) not null,
进程已结束，退出代码 0
```

(2) 拼音搜索单元测试

```
//测试

public static void main(String[] args) {

    System.out.println("1998".matches(".*[a-z].*")); // 字符匹配是字母

    System.out.println(Arrays.toString(get("我爱我家"))); // 纯中文字符

    System.out.println(Arrays.toString(get("我 love 我家")));

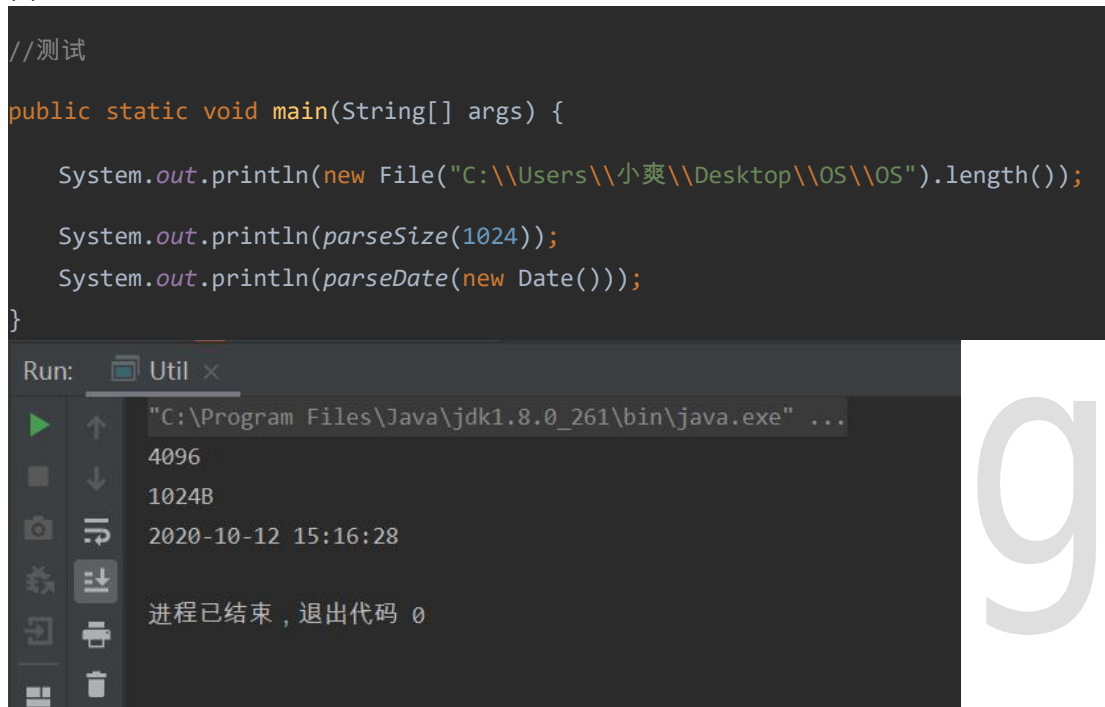
    System.out.println(Arrays.toString(
        compose(get("干一行", true)))); // 全拼

    System.out.println(Arrays.toString(
        compose(get("长长", false)))); // 首拼
}
```




```
Run: PinyinUtil x
"C:\Program Files\Java\jdk1.8.0_261\bin\java.exe" ...
false
[woaiwojia, wawj]
[wolvewojia, wlovewj]
[ganyixing, ganyihang, ganyiheng]
[cc, cz, zc, zz]
进程已结束, 退出代码 0
```

(3) 文件格式解析单元测试



```
//测试
public static void main(String[] args) {
    System.out.println(new File("C:\\Users\\小爽\\Desktop\\OS\\OS").length());
    System.out.println(parseSize(1024));
    System.out.println(parseDate(new Date()));
}
Run: Util x
"C:\Program Files\Java\jdk1.8.0_261\bin\java.exe" ...
4096
10248
2020-10-12 15:16:28
进程已结束, 退出代码 0
```

- 2、界面测试
- 3、功能测试
- 4、性能测试

五、项目总结

此项目利用多线程提高了文件扫描效率，同时使用拼音工具类实现了文件全拼和首拼搜索，提高了文件搜索效率。

改进之处：搜索到指定文件后，可以链接打开此文件。