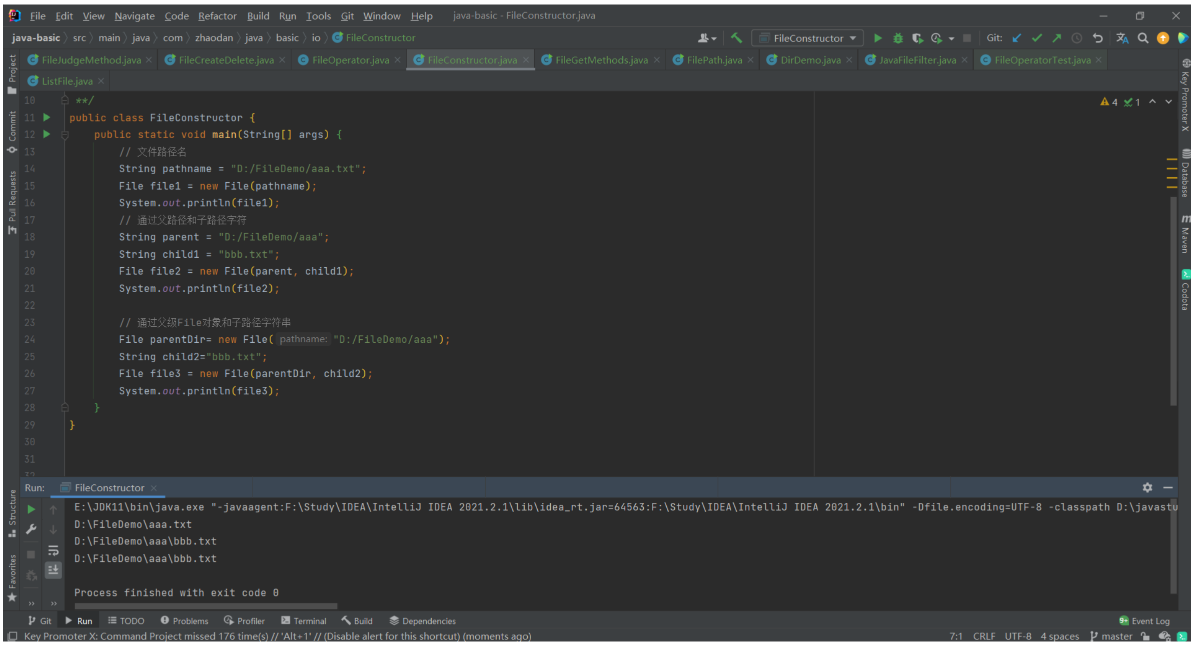
IO学习

1.1File类的构造方法

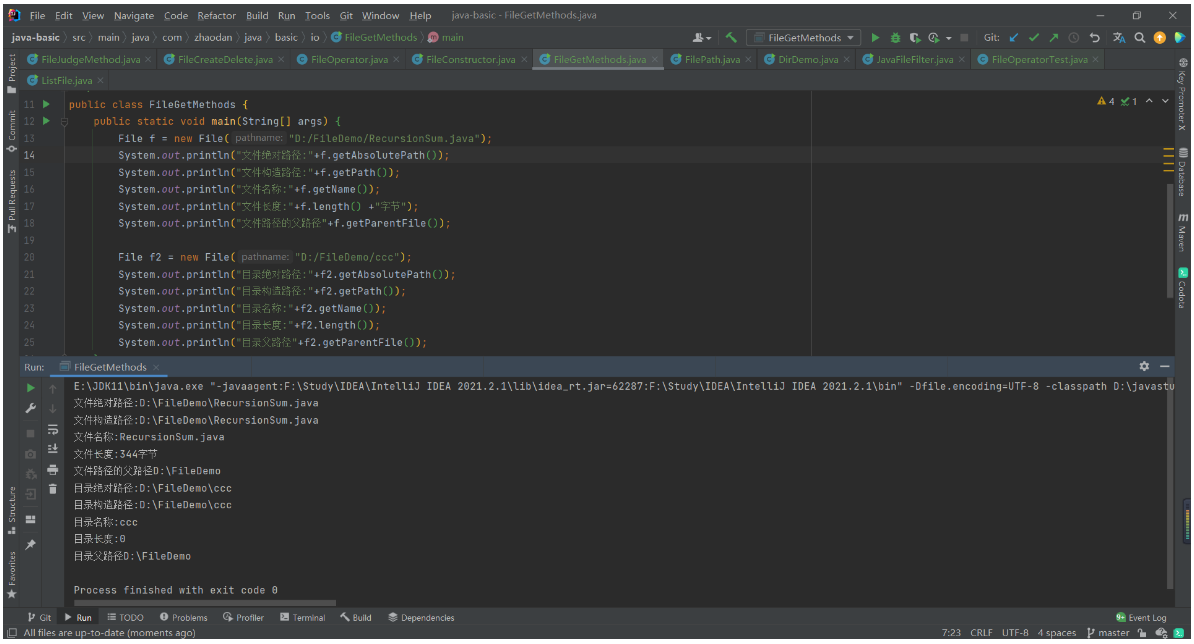


```java
	**/
public class FileConstructor {
    public static void main(String[] args) {
        // 文件路径名
        String pathname = "D:/FileDemo/aaa.txt";
        File file1 = new File(pathname);
        System.out.println(file1);
        // 通过父路径和子路径字符
        String parent = "D:/FileDemo/aaa";
        String child1 = "bbb.txt";
        File file2 = new File(parent, child1);
        System.out.println(file2);

        // 通过父级File对象和子路径字符串
        File parentDir= new File("D:/FileDemo/aaa");
        String child2="bbb.txt";
        File file3 = new File(parentDir, child2);
        System.out.println(file3);
    }
}
```

```
E:\JDK11\bin\java.exe "-javaagent:F:\Study\IDEA\IntelliJ IDEA 2021.2.1\lib\idea_rt.jar=64563:F:\Study\IDEA\IntelliJ IDEA 2021.2.1\bin" -Dfile.encoding=UTF-8 -classpath D:\javastu
D:\FileDemo\aaa.txt
D:\FileDemo\aaa\bbb.txt
D:\FileDemo\aaa\bbb.txt

Process finished with exit code 0
```

Key Promoter X: Command Project missed 176 time(s) // 'Alt+1' // (Disable alert for this shortcut) (moments ago)

1.2File类的获取方法

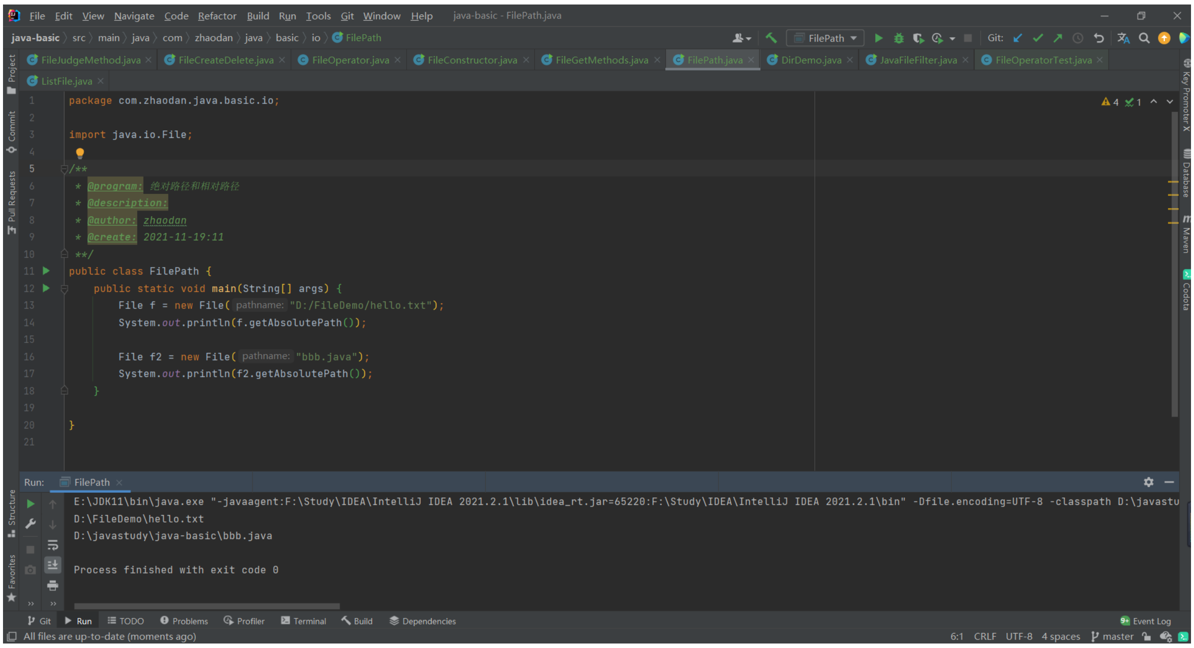


```java
public class FileGetMethods {
    public static void main(String[] args) {
        File f = new File("D:/FileDemo/RecursionSum.java");
        System.out.println("文件绝对路径:"+f.getAbsolutePath());
        System.out.println("文件构造路径:"+f.getPath());
        System.out.println("文件名称:"+f.getName());
        System.out.println("文件长度:"+f.length() +"字节");
        System.out.println("文件路径的父路径"+f.getParentFile());

        File f2 = new File("D:/FileDemo/ccc");
        System.out.println("目录绝对路径:"+f2.getAbsolutePath());
        System.out.println("目录构造路径:"+f2.getPath());
        System.out.println("目录名称:"+f2.getName());
        System.out.println("目录长度:"+f2.length());
        System.out.println("目录父路径"+f2.getParentFile());
```

```
E:\JDK11\bin\java.exe "-javaagent:F:\Study\IDEA\IntelliJ IDEA 2021.2.1\lib\idea_rt.jar=62287:F:\Study\IDEA\IntelliJ IDEA 2021.2.1\bin" -Dfile.encoding=UTF-8 -classpath D:\javastu
文件绝对路径:D:\FileDemo\RecursionSum.java
文件构造路径:D:\FileDemo\RecursionSum.java
文件名称:RecursionSum.java
文件长度:344字节
文件路径的父路径D:\FileDemo
目录绝对路径:D:\FileDemo\ccc
目录构造路径:D:\FileDemo\ccc
目录名称:ccc
目录长度:0
目录父路径D:\FileDemo

Process finished with exit code 0
```

All files are up-to-date (moments ago)

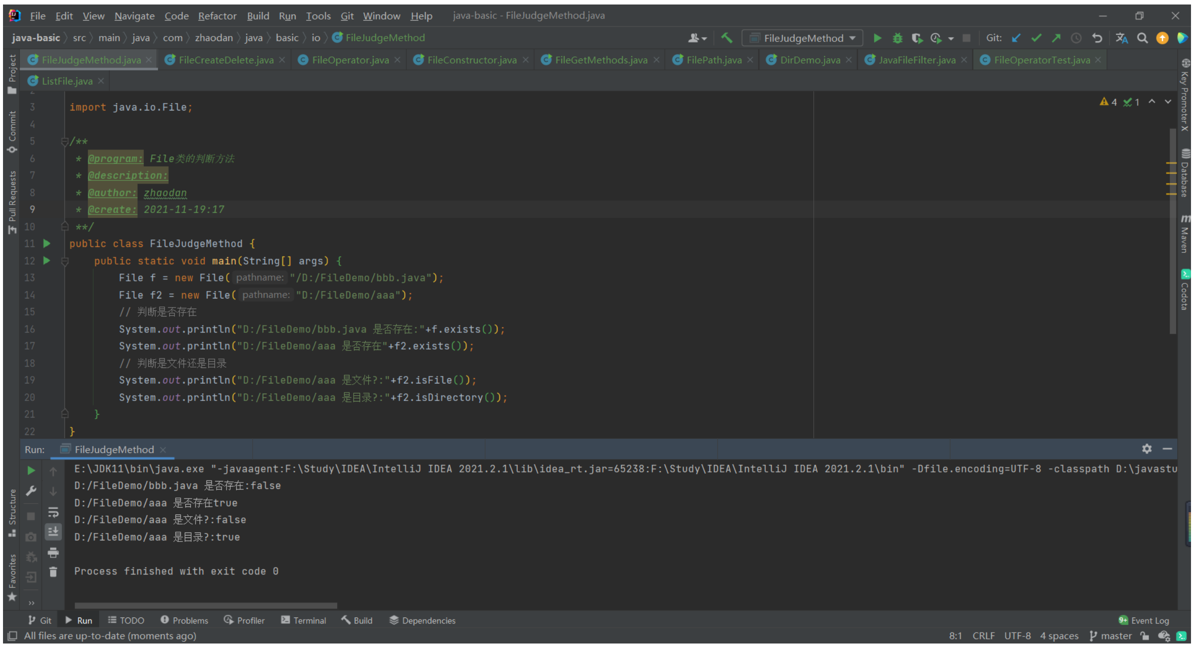
1.3绝对路径和相对路径



```java
package com.zhaodan.java.basic.io;

import java.io.File;

/**
 * @program: 绝对路径和相对路径
 * @description:
 * @author: zhaodan
 * @create: 2021-11-19:11
 **/
public class FilePath {
    public static void main(String[] args) {
        File f = new File( pathname: "D:/FileDemo/hello.txt");
        System.out.println(f.getAbsolutePath());

        File f2 = new File( pathname: "bbb.java");
        System.out.println(f2.getAbsolutePath());
    }

}
```

```
E:\JDK11\bin\java.exe "-javaagent:F:\Study\IDEA\IntelliJ IDEA 2021.2.1\lib\idea_rt.jar=65220:F:\Study\IDEA\IntelliJ IDEA 2021.2.1\bin" -Dfile.encoding=UTF-8 -classpath D:\javastu
D:\FileDemo\hello.txt
D:\javastudy\java-basic\bbb.java

Process finished with exit code 0
```

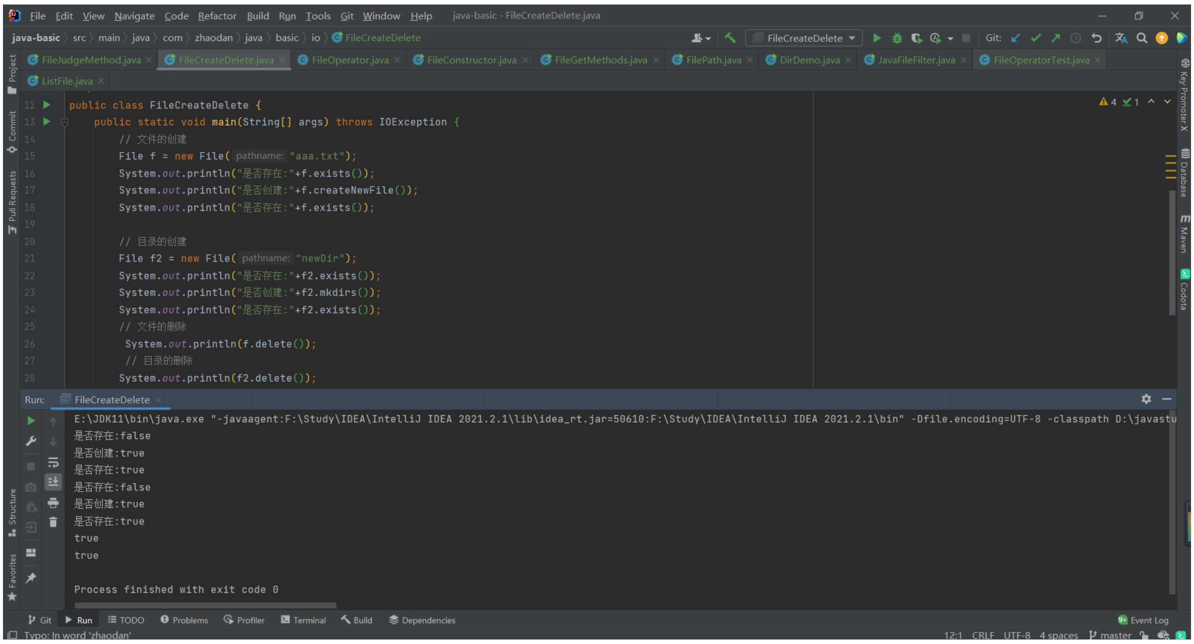
1.4File类判断方法


java-basic - FileJudgeMethod.java
java-basic 〉 src 〉 main 〉 java 〉 com 〉 zhaodan 〉 java 〉 basic 〉 io 〉 FileJudgeMethod
Run: FileJudgeMethod

```java
import java.io.File;

/**
 * @program: File类的判断方法
 * @description:
 * @author: zhaodan
 * @create: 2021-11-19:17
 **/
public class FileJudgeMethod {
    public static void main(String[] args) {
        File f = new File( pathname: "/D:/FileDemo/bbb.java");
        File f2 = new File( pathname: "D:/FileDemo/aaa");
        // 判断是否存在
        System.out.println("D:/FileDemo/bbb.java 是否存在:"+f.exists());
        System.out.println("D:/FileDemo/aaa 是否存在"+f2.exists());
        // 判断是文件还是目录
        System.out.println("D:/FileDemo/aaa 是文件?:"+f2.isFile());
        System.out.println("D:/FileDemo/aaa 是目录?:"+f2.isDirectory());
    }
}
```

```
E:\JDK11\bin\java.exe "-javaagent:F:\Study\IDEA\IntelliJ IDEA 2021.2.1\lib\idea_rt.jar=65238:F:\Study\IDEA\IntelliJ IDEA 2021.2.1\bin" -Dfile.encoding=UTF-8 -classpath D:\javastu
D:/FileDemo/bbb.java 是否存在:false
D:/FileDemo/aaa 是否存在true
D:/FileDemo/aaa 是文件?:false
D:/FileDemo/aaa 是目录?:true

Process finished with exit code 0
```

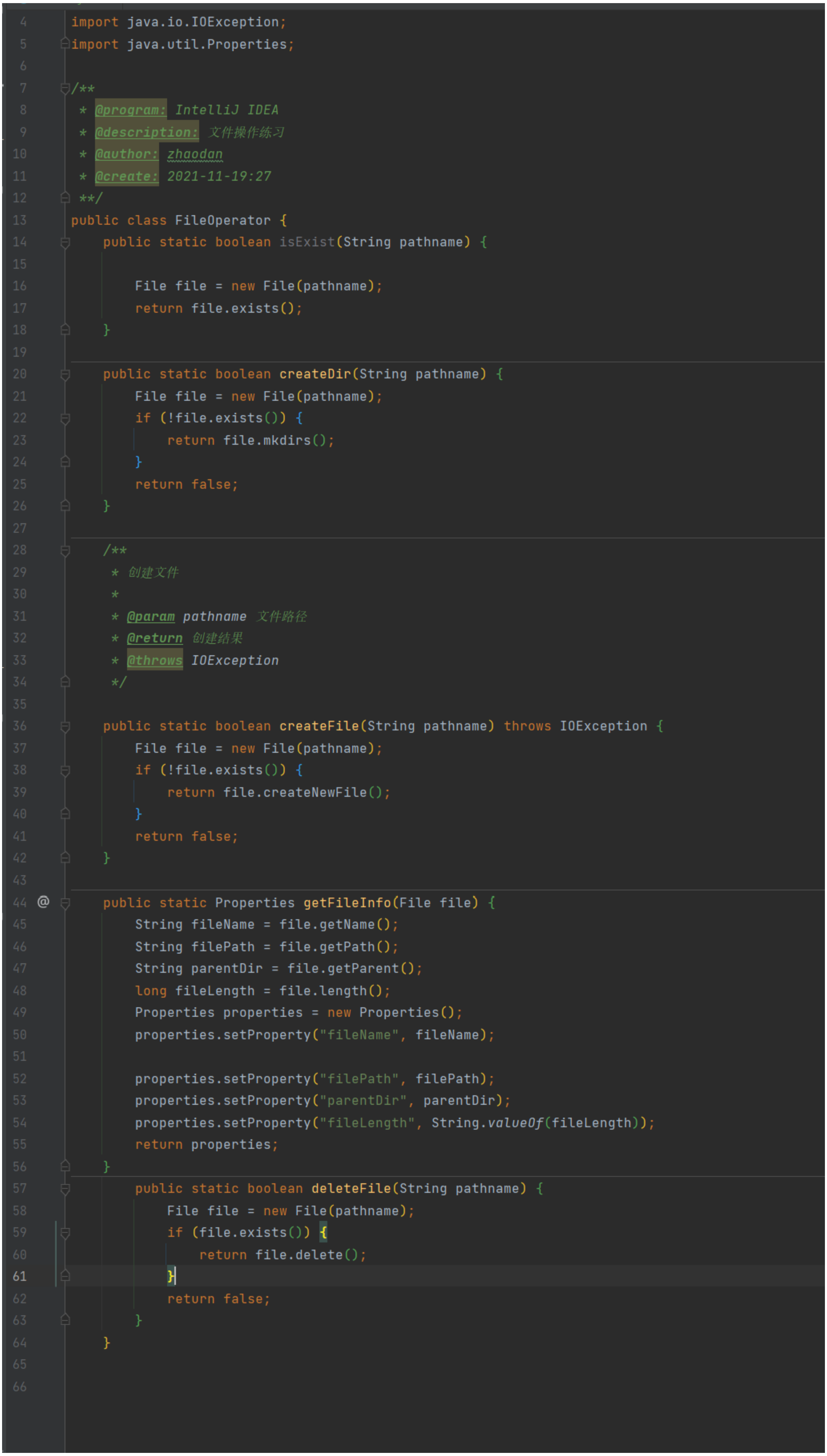
1.5File类创建删除功能的方法



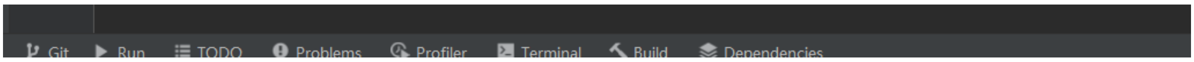
```java
public class FileCreateDelete {
    public static void main(String[] args) throws IOException {
        // 文件的创建
        File f = new File( pathname: "aaa.txt");
        System.out.println("是否存在:"+f.exists());
        System.out.println("是否创建:"+f.createNewFile());
        System.out.println("是否存在:"+f.exists());

        // 目录的创建
        File f2 = new File( pathname: "newDir");
        System.out.println("是否存在:"+f2.exists());
        System.out.println("是否创建:"+f2.mkdirs());
        System.out.println("是否存在:"+f2.exists());
        // 文件的删除
         System.out.println(f.delete());
         // 目录的删除
        System.out.println(f2.delete());
```

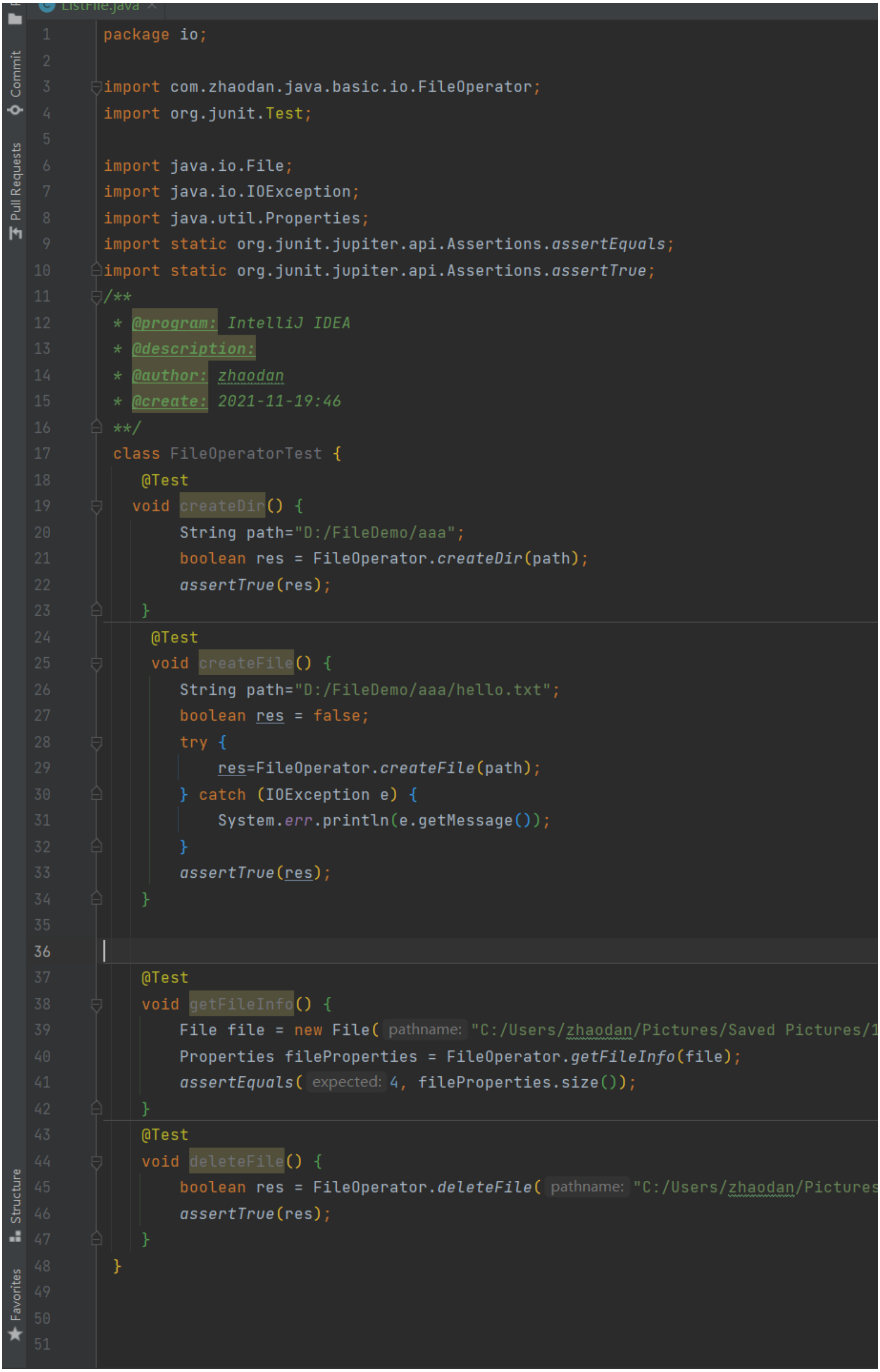
```
E:\JDK11\bin\java.exe "-javaagent:F:\Study\IDEA\IntelliJ IDEA 2021.2.1\lib\idea_rt.jar=50610:F:\Study\IDEA\IntelliJ IDEA 2021.2.1\bin" -Dfile.encoding=UTF-8 -classpath D:\javastu
是否存在:false
是否创建:true
是否存在:true
是否存在:false
是否创建:true
是否存在:true
true
true

Process finished with exit code 0
```

1.6File综合操作



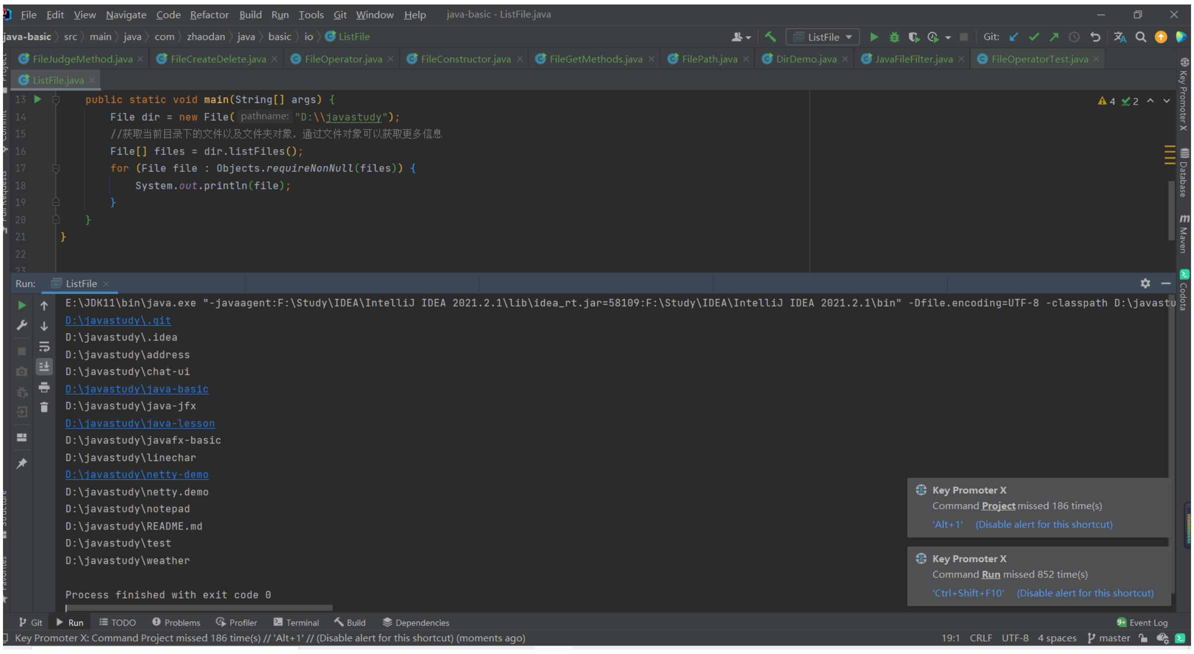
```java
import java.io.IOException;
import java.util.Properties;

/**
 * @program: IntelliJ IDEA
 * @description: 文件操作练习
 * @author: zhaodan
 * @create: 2021-11-19:27
 **/
public class FileOperator {
    public static boolean isExist(String pathname) {

        File file = new File(pathname);
        return file.exists();
    }

    public static boolean createDir(String pathname) {
        File file = new File(pathname);
        if (!file.exists()) {
            return file.mkdirs();
        }
        return false;
    }

    /**
     * 创建文件
     *
     * @param pathname 文件路径
     * @return 创建结果
     * @throws IOException
     */

    public static boolean createFile(String pathname) throws IOException {
        File file = new File(pathname);
        if (!file.exists()) {
            return file.createNewFile();
        }
        return false;
    }

    public static Properties getFileInfo(File file) {
        String fileName = file.getName();
        String filePath = file.getPath();
        String parentDir = file.getParent();
        long fileLength = file.length();
        Properties properties = new Properties();
        properties.setProperty("fileName", fileName);

        properties.setProperty("filePath", filePath);
        properties.setProperty("parentDir", parentDir);
        properties.setProperty("fileLength", String.valueOf(fileLength));
        return properties;
    }
        public static boolean deleteFile(String pathname) {
            File file = new File(pathname);
            if (file.exists()) {
                return file.delete();
            }
            return false;
        }
    }


```





```java
package io;

import com.zhaodan.java.basic.io.FileOperator;
import org.junit.Test;

import java.io.File;
import java.io.IOException;
import java.util.Properties;
import static org.junit.jupiter.api.Assertions.assertEquals;
import static org.junit.jupiter.api.Assertions.assertTrue;
/**
 * @program: IntelliJ IDEA
 * @description:
 * @author: zhaodan
 * @create: 2021-11-19:46
 **/
 class FileOperatorTest {
    @Test
   void createDir() {
        String path="D:/FileDemo/aaa";
        boolean res = FileOperator.createDir(path);
        assertTrue(res);
    }
     @Test
     void createFile() {
        String path="D:/FileDemo/aaa/hello.txt";
        boolean res = false;
        try {
            res=FileOperator.createFile(path);
        } catch (IOException e) {
            System.err.println(e.getMessage());
        }
        assertTrue(res);
    }


    @Test
    void getFileInfo() {
        File file = new File( pathname: "C:/Users/zhaodan/Pictures/Saved Pictures/1
        Properties fileProperties = FileOperator.getFileInfo(file);
        assertEquals( expected: 4, fileProperties.size());
    }
    @Test
    void deleteFile() {
        boolean res = FileOperator.deleteFile( pathname: "C:/Users/zhaodan/Pictures
        assertTrue(res);
    }
 }
```

1.7File类目录遍历方法



```java
    public static void main(String[] args) {
        File dir = new File( pathname: "D:\\javastudy");
        //获取当前目录下的文件以及文件夹对象，通过文件对象可以获取更多信息
        File[] files = dir.listFiles();
        for (File file : Objects.requireNonNull(files)) {
            System.out.println(file);
        }
    }
}
```

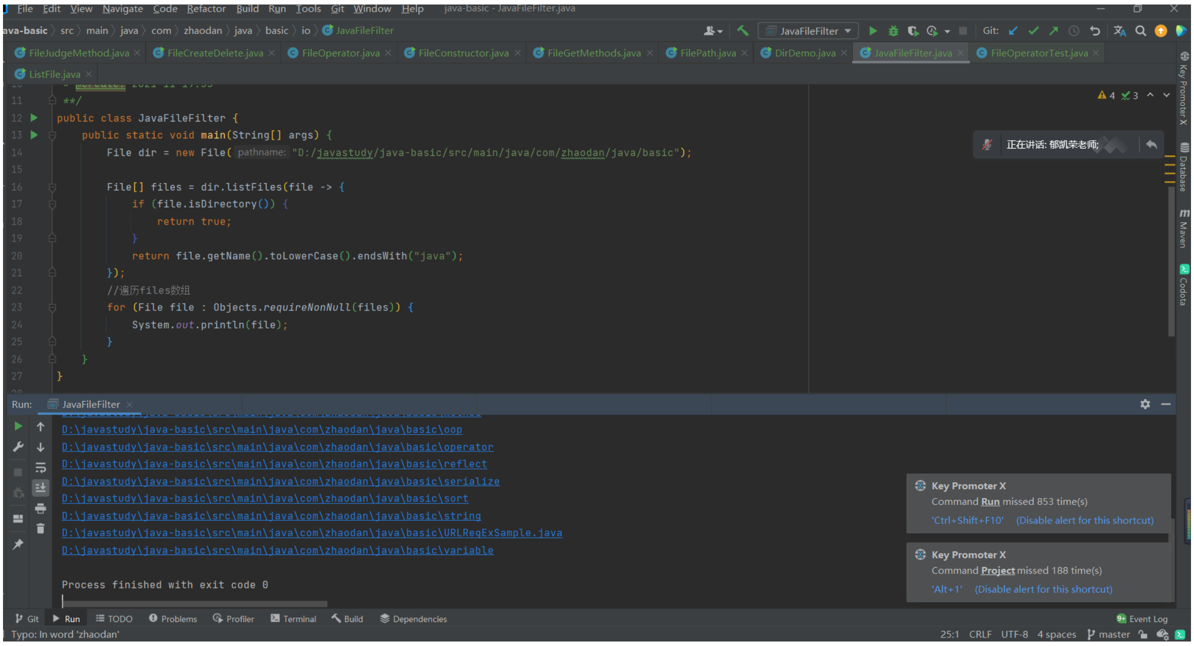
```
E:\JDK11\bin\java.exe "-javaagent:F:\Study\IDEA\IntelliJ IDEA 2021.2.1\lib\idea_rt.jar=58109:F:\Study\IDEA\IntelliJ IDEA 2021.2.1\bin" -Dfile.encoding=UTF-8 -classpath D:\javastu
D:\javastudy\.git
D:\javastudy\.idea
D:\javastudy\address
D:\javastudy\chat-ui
D:\javastudy\java-basic
D:\javastudy\java-jfx
D:\javastudy\java-lesson
D:\javastudy\javafx-basic
D:\javastudy\linechar
D:\javastudy\netty-demo
D:\javastudy\netty.demo
D:\javastudy\notepad
D:\javastudy\README.md
D:\javastudy\test
D:\javastudy\weather

Process finished with exit code 0
```

Key Promoter X
Command Project missed 186 time(s)
'Alt+1' (Disable alert for this shortcut)

Key Promoter X
Command Run missed 852 time(s)
'Ctrl+Shift+F10' (Disable alert for this shortcut)

Key Promoter X: Command Project missed 186 time(s) // 'Alt+1' // (Disable alert for this shortcut) (moments ago)

1.8FileFilter接口



```java
 **/
public class JavaFileFilter {
    public static void main(String[] args) {
        File dir = new File( pathname: "D:/javastudy/java-basic/src/main/java/com/zhaodan/java/basic");

        File[] files = dir.listFiles(file -> {
            if (file.isDirectory()) {
                return true;
            }
            return file.getName().toLowerCase().endsWith("java");
        });
        //遍历files数组
        for (File file : Objects.requireNonNull(files)) {
            System.out.println(file);
        }
    }
}
```

正在讲话: 郁凯荣老师;

```
D:\javastudy\java-basic\src\main\java\com\zhaodan\java\basic\oop
D:\javastudy\java-basic\src\main\java\com\zhaodan\java\basic\operator
D:\javastudy\java-basic\src\main\java\com\zhaodan\java\basic\reflect
D:\javastudy\java-basic\src\main\java\com\zhaodan\java\basic\serialize
D:\javastudy\java-basic\src\main\java\com\zhaodan\java\basic\sort
D:\javastudy\java-basic\src\main\java\com\zhaodan\java\basic\string
D:\javastudy\java-basic\src\main\java\com\zhaodan\java\basic\URLRegExSample.java
D:\javastudy\java-basic\src\main\java\com\zhaodan\java\basic\variable

Process finished with exit code 0
```

Key Promoter X
Command Run missed 853 time(s)
'Ctrl+Shift+F10' (Disable alert for this shortcut)

Key Promoter X
Command Project missed 188 time(s)
'Alt+1' (Disable alert for this shortcut)

Typo: In word 'zhaodan'

2.1 概述

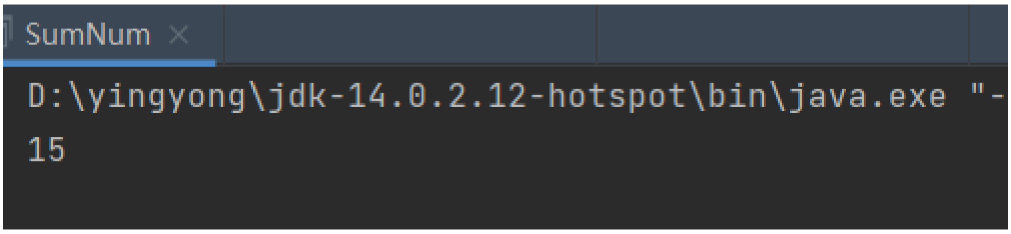
递归：指在当前方法内调用自己的这种现象

```java
public static void a(){ a();
}
```

2.2 递归求和

计算1 ~ n**的和**

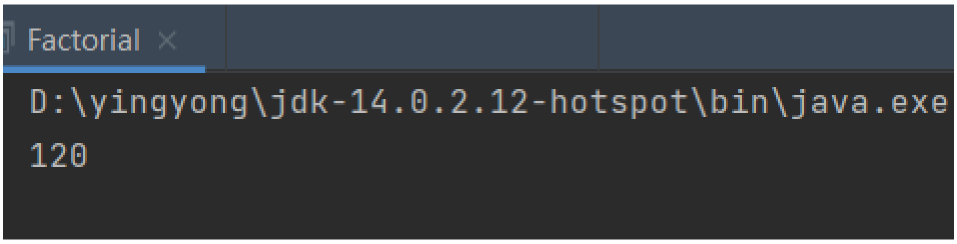
```java
package homework.递归;

/**
 * @program: learn
 * @description:
 * @author: zwx
 * @create: 2021-11-17 12:45
 **/
public class SumNum {
    public static void main(String[] args) {
        int num=5;
        int sum = SUM.getSum(num);
        System.out.println(sum);
    }
}
class SUM{
    public static int getSum(int num){
        if( num ==1){
            return 1;
        }return num+getSum(num-1);
    }
}
```



```
D:\yingyong\jdk-14.0.2.12-hotspot\bin\java.exe "-
15
```

2.3 递归求阶乘

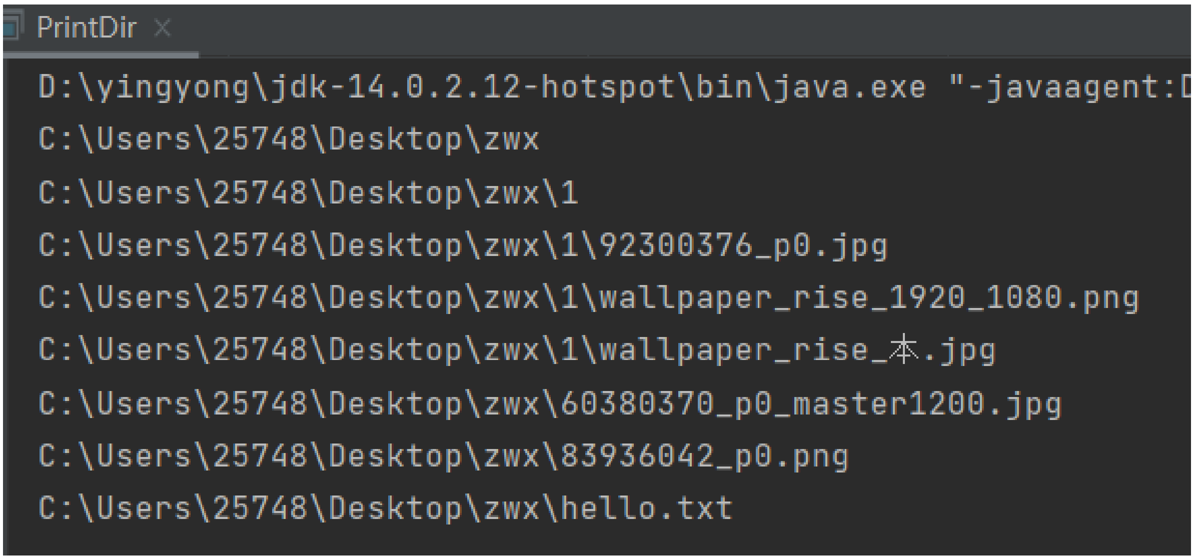
```
package homework.递归;

/**
 * @program: learn
 * @description:
 * @author: zwx
 * @create: 2021-11-17 12:48
 **/
public class Factorial {
    public static void main(String[] args) {
        System.out.println(T.factorial(5));
    }
}
class T {
    public static int factorial(int num) {
        if (num == 1) {
            return 1;
        } else {
            return num * factorial(num - 1);
        }
    }
}
```



```
D:\yingyong\jdk-14.0.2.12-hotspot\bin\java.exe
120
```

2.4 目录遍历

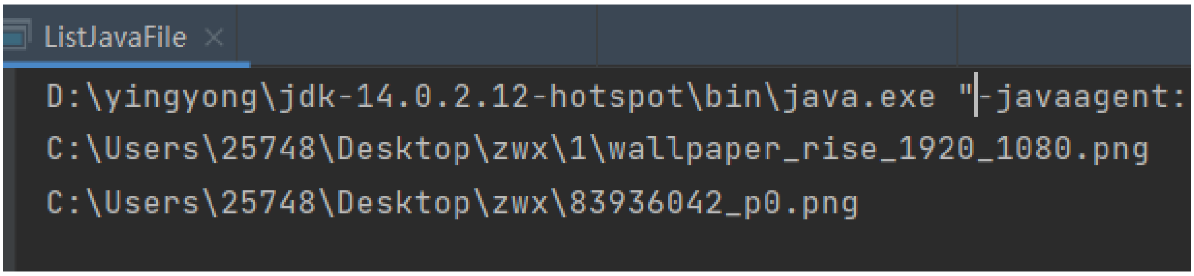
```java
package homework.递归;

import java.io.File;
import java.util.Objects;

/**
 * @program: learn
 * @description:
 * @author: zwx
 * @create: 2021-11-17 13:01
 **/
public class PrintDir {
    public static void main(String[] args) {
        // 创建File对象
        File dir = new File("C:/Users/25748/Desktop/zwx");
        // 调用打印目录方法
        printDir(dir);
    }

    public static void printDir(File dir) {
        System.out.println(dir);
        // 获取子文件和目录
        File[] files = dir.listFiles();
        // 循环打印
        for (File file : Objects.requireNonNull(files)) {
            //判断是文件，直接输出
            if (file.isFile()) {
                System.out.println(file);
            } else {
                // 是目录，继续遍历,形成递归
                printDir(file);
            }
        }
    }
}
```



```
D:\yingyong\jdk-14.0.2.12-hotspot\bin\java.exe "-javaagent:
C:\Users\25748\Desktop\zwx
C:\Users\25748\Desktop\zwx\1
C:\Users\25748\Desktop\zwx\1\92300376_p0.jpg
C:\Users\25748\Desktop\zwx\1\wallpaper_rise_1920_1080.png
C:\Users\25748\Desktop\zwx\1\wallpaper_rise_本.jpg
C:\Users\25748\Desktop\zwx\60380370_p0_master1200.jpg
C:\Users\25748\Desktop\zwx\83936042_p0.png
C:\Users\25748\Desktop\zwx\hello.txt
```

2.5 目录遍历搜索png文件

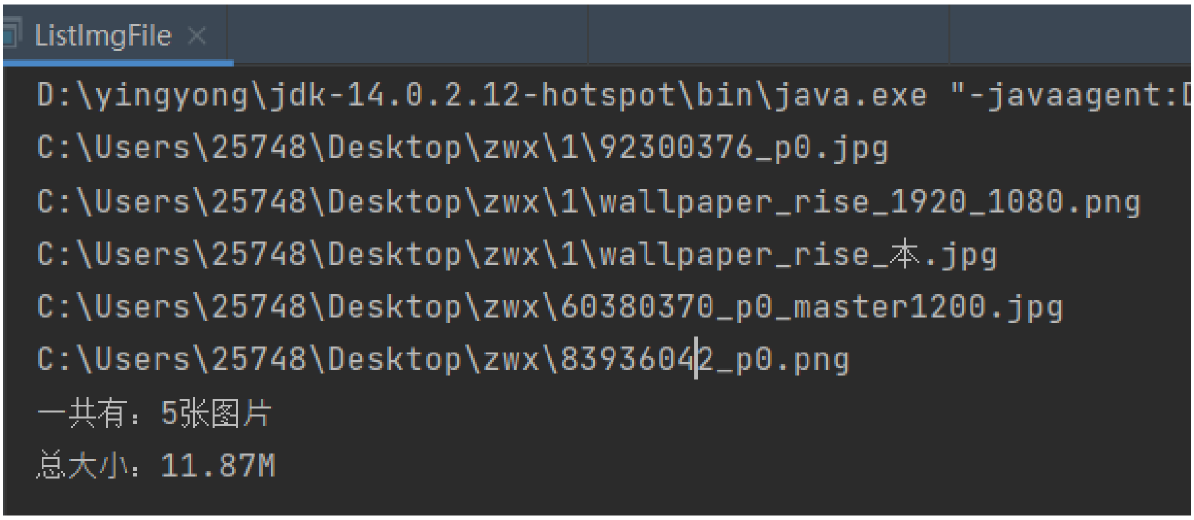
```java
package homework.递归;

import java.io.File;
import java.util.Objects;

/**
 * @program: learn
 * @description:
 * @author: zwx
 * @create: 2021-11-17 13:06
 **/
public class ListJavaFile {
    public static void main(String[] args) {
        // 创建File对象
        File dir = new File("C:/Users/25748/Desktop/zwx");
        // 调用打印目录方法
        printDir(dir);
    }

    public static void printDir(File dir) {
        // 获取子文件和目录
        File[] files = dir.listFiles(pathname -> {
            if (pathname.isDirectory()) {
                return true;
            }
            return pathname.getName().toLowerCase().endsWith("png");
        });
        // 循环打印
        for (File file : Objects.requireNonNull(files)) {
            if (file.isFile()) {
                System.out.println(file);
            } else {
                // 是目录，继续遍历,形成递归
                printDir(file);
            }
        }
    }
}
```



```
D:\yingyong\jdk-14.0.2.12-hotspot\bin\java.exe "-javaagent:
C:\Users\25748\Desktop\zwx\1\wallpaper_rise_1920_1080.png
C:\Users\25748\Desktop\zwx\83936042_p0.png
```

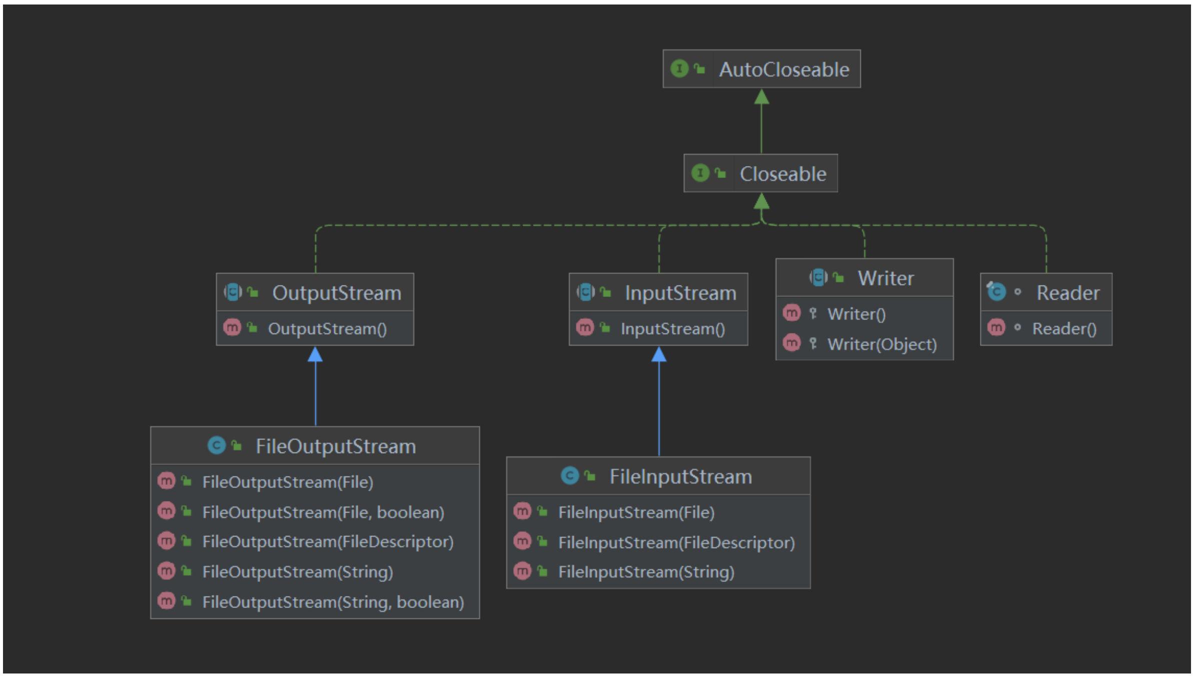
2.6 目录遍历搜索图片并统计

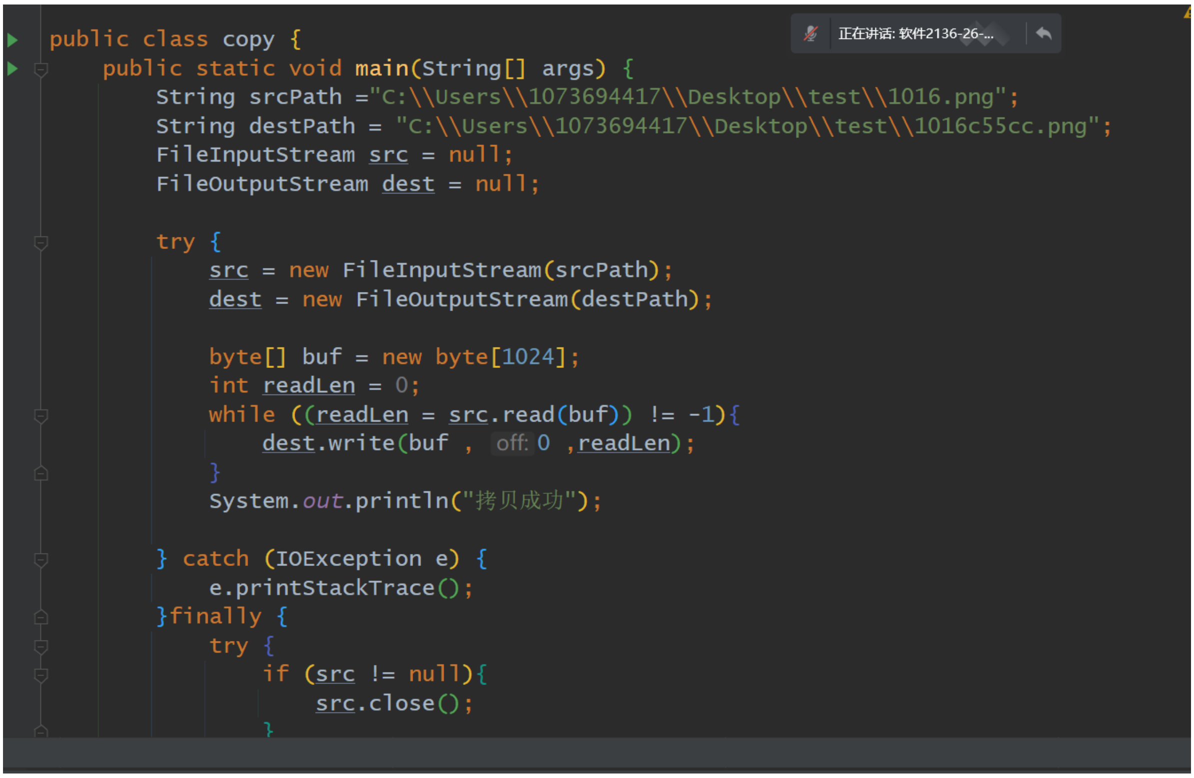
```java
package homework.递归;

import java.io.File;
import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.List;
import java.util.Objects;

/**
 * @program: learn
 * @description:
 * @author: zwx
 * @create: 2021-11-17 13:09
 **/
public class ListImgFile {
    private static List<File> fileList = new ArrayList<>();
    private static long size;
    private static final double UNIT = 1024.0;

    public static void main(String[] args) {
        //创建File对象
        File dir = new File("C:/Users/25748/Desktop/zwx");
        //调用遍历目录的方法，得到返回集合
        fileList = listFolder(dir);
        //遍历集合
        fileList.forEach(System.out::println);
        //总数
        System.out.println("一共有：" + fileList.size() + "张图片");
        //大小（计算转换为KB、MB、GB等）
        System.out.println("总大小：" + fileSize(size));
    }


    /**
     * 递归方法，遍历指定目录，找出所有的图片文件
     *
     * @param dir 目录
     * @return 图片文件集合
     */
    private static List<File> listFolder(File dir) {
        //使用文件过滤器过滤图片文件
        File[] files = dir.listFiles(file -> {
            //是目录就放行
            if (file.isDirectory()) {
                return true;
            }
            //是图片类型文件也放行
            return isImage(file);
        });

        //遍历files数组
        for (File file : Objects.requireNonNull(files)) {
            //如果file是文件
            if (file.isFile()) {
                //将当前file的长度累加到size
                size += file.length();
                //将当前文件加入集合
                fileList.add(file);
            } else {
                //如果file是目录，递归遍历
                listFolder(file);
            }
        }
        return fileList;
    }


    /**
     * 判断file是否为图片类型文件
     * @return 判断结果
     * @param file 文件对象
     */
    private static boolean isImage(File file) {
        String fileName = file.getName().toLowerCase();
        return fileName.endsWith("jpg") || fileName.endsWith("jpeg")
                || fileName.endsWith("png") || fileName.endsWith("gif")
                || fileName.endsWith("bmp") || fileName.endsWith("webp");
    }


    /**
     * 根据字节数换算文件大小
     *
     * @param length 字节数
     * @return String
     */
    private static String fileSize(long length) {
        DecimalFormat df = new DecimalFormat("#.00");
        double res = length / UNIT;
        if (res < 1) {
            return length + "字节";
        } else if (res < UNIT) {
            return df.format(res) + "K";
        } else if (res < UNIT * UNIT) {
            return df.format(res / UNIT) + "M";
        } else {
            return df.format(res / UNIT / UNIT) + "G";
        }
    }
}
```



```
D:\yingyong\jdk-14.0.2.12-hotspot\bin\java.exe "-javaagent:D
C:\Users\25748\Desktop\zwx\1\92300376_p0.jpg
C:\Users\25748\Desktop\zwx\1\wallpaper_rise_1920_1080.png
C:\Users\25748\Desktop\zwx\1\wallpaper_rise_本.jpg
C:\Users\25748\Desktop\zwx\60380370_p0_master1200.jpg
C:\Users\25748\Desktop\zwx\83936042_p0.png
一共有：5张图片
总大小：11.87M
```

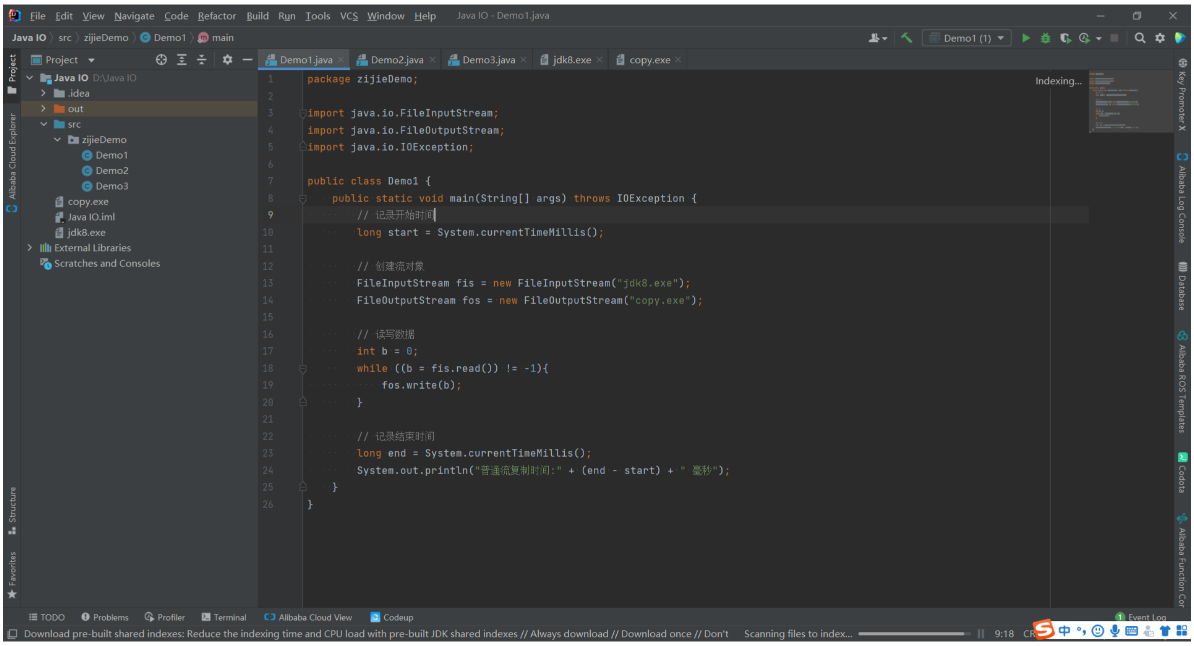
3.1





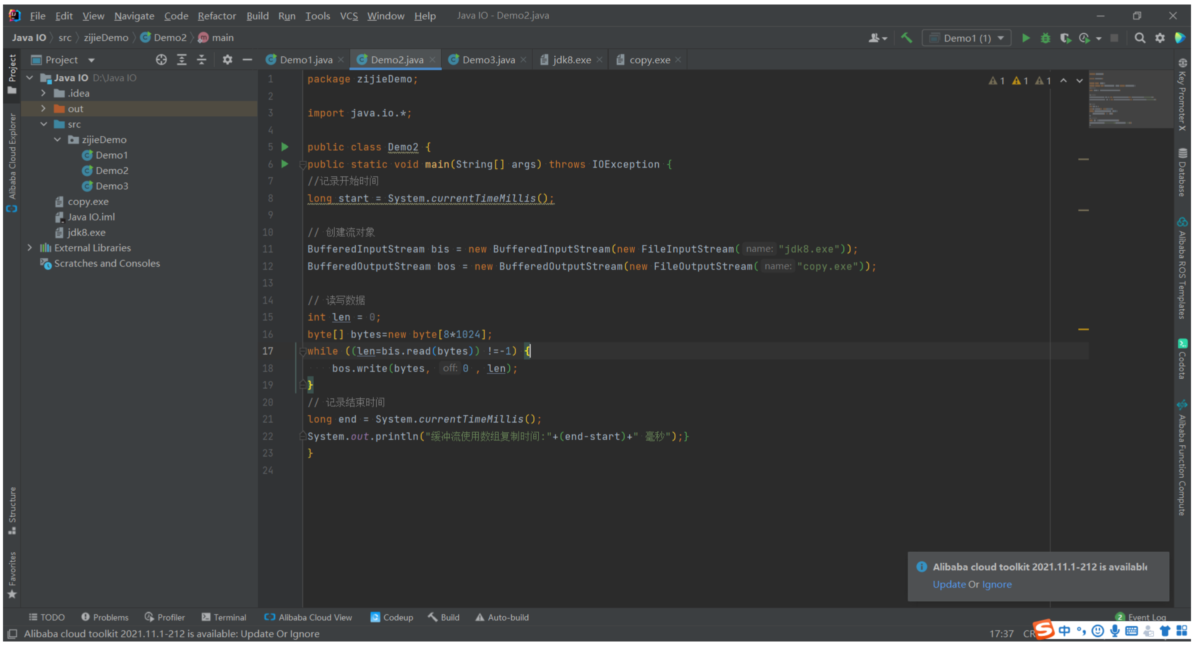
```java
public class copy {
    public static void main(String[] args) {
        String srcPath ="C:\\Users\\1073694417\\Desktop\\test\\1016.png";
        String destPath = "C:\\Users\\1073694417\\Desktop\\test\\1016c55cc.png";
        FileInputStream src = null;
        FileOutputStream dest = null;

        try {
            src = new FileInputStream(srcPath);
            dest = new FileOutputStream(destPath);

            byte[] buf = new byte[1024];
            int readLen = 0;
            while ((readLen = src.read(buf)) != -1){
                dest.write(buf , off: 0 ,readLen);
            }
            System.out.println("拷贝成功");

        } catch (IOException e) {
            e.printStackTrace();
        }finally {
            try {
                if (src != null){
                    src.close();
                }
```

4.



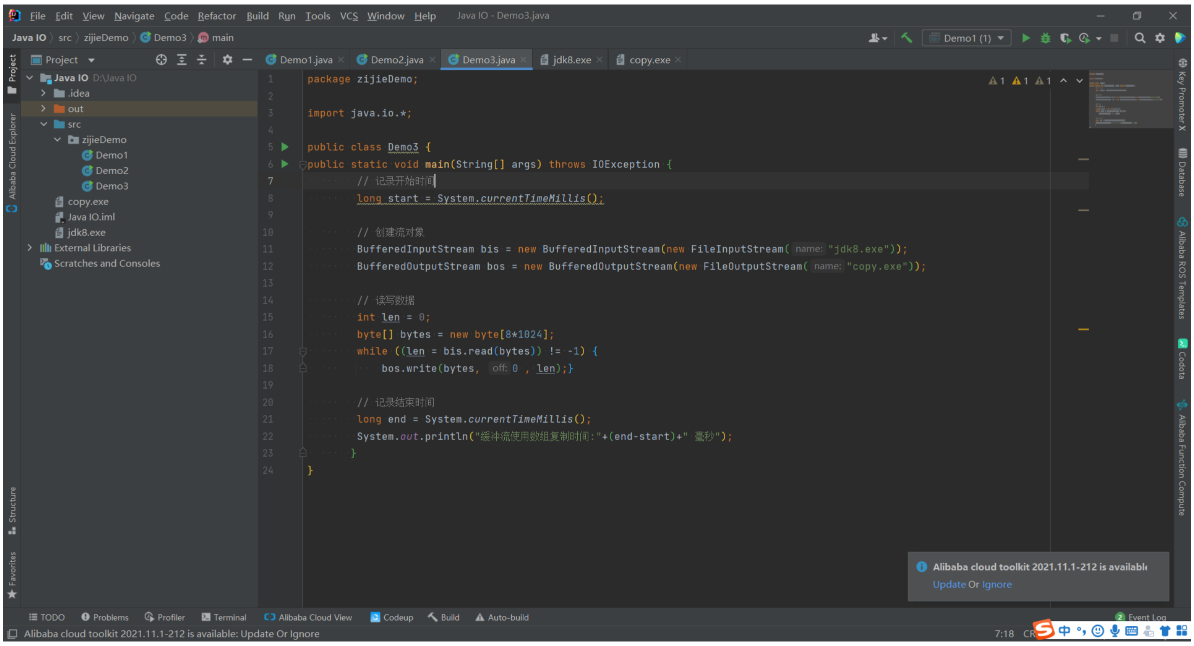
```java
package zijieDemo;

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class Demo1 {
    public static void main(String[] args) throws IOException {
        // 记录开始时间
        long start = System.currentTimeMillis();

        // 创建流对象
        FileInputStream fis = new FileInputStream("jdk8.exe");
        FileOutputStream fos = new FileOutputStream("copy.exe");

        // 读写数据
        int b = 0;
        while ((b = fis.read()) != -1){
            fos.write(b);
        }

        // 记录结束时间
        long end = System.currentTimeMillis();
        System.out.println("普通流复制时间:" + (end - start) + " 毫秒");
    }
}
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK shared indexes // Always download // Download once // Don't



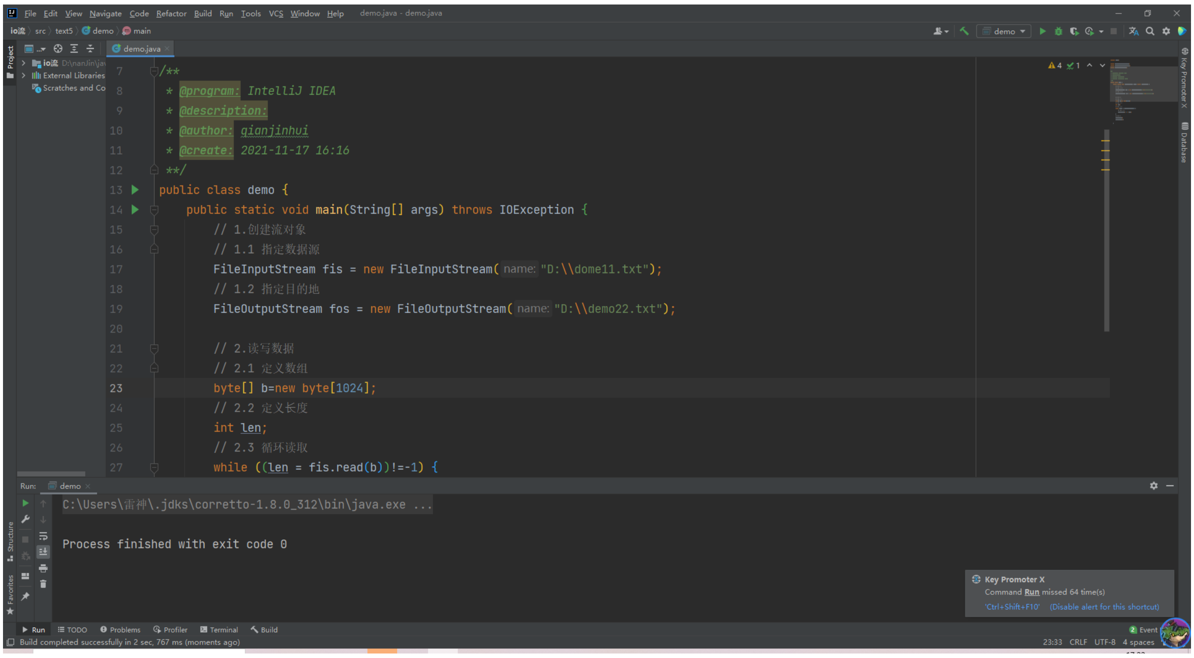
```java
package zijieDemo;

import java.io.*;

public class Demo2 {
public static void main(String[] args) throws IOException {
//记录开始时间
long start = System.currentTimeMillis();

// 创建流对象
BufferedInputStream bis = new BufferedInputStream(new FileInputStream("jdk8.exe"));
BufferedOutputStream bos = new BufferedOutputStream(new FileOutputStream("copy.exe"));

// 读写数据
int len = 0;
byte[] bytes=new byte[8*1024];
while ((len=bis.read(bytes)) !=-1) {
    bos.write(bytes, 0 , len);
}
// 记录结束时间
long end = System.currentTimeMillis();
System.out.println("缓冲流使用数组复制时间:"+(end-start)+" 毫秒");}
}
```

Alibaba cloud toolkit 2021.11.1-212 is available: Update Or Ignore



```java
package zijieDemo;

import java.io.*;

public class Demo3 {
public static void main(String[] args) throws IOException {
        // 记录开始时间
        long start = System.currentTimeMillis();

        // 创建流对象
        BufferedInputStream bis = new BufferedInputStream(new FileInputStream("jdk8.exe"));
        BufferedOutputStream bos = new BufferedOutputStream(new FileOutputStream("copy.exe"));

        // 读写数据
        int len = 0;
        byte[] bytes = new byte[8*1024];
        while ((len = bis.read(bytes)) != -1) {
            bos.write(bytes, 0 , len);}

        // 记录结束时间
        long end = System.currentTimeMillis();
        System.out.println("缓冲流使用数组复制时间:"+(end-start)+" 毫秒");
        }
}
```

Alibaba cloud toolkit 2021.11.1-212 is available: Update Or Ignore


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
demo.java - demo.java
io流 〉 src 〉 text5 〉 demo 〉 main
Project
io流 D:\nanJin\ja
External Libraries
Scratches and Co
demo.java
Structure
Favorites
Run: demo
Run TODO Problems Profiler Terminal Build
Event
23:33 CRLF UTF-8 4 spaces

```
/**
 * @program: IntelliJ IDEA
 * @description:
 * @author: qianjinhui
 * @create: 2021-11-17 16:16
 **/
public class demo {
    public static void main(String[] args) throws IOException {
        // 1.创建流对象
        // 1.1 指定数据源
        FileInputStream fis = new FileInputStream( name: "D:\\dome11.txt");
        // 1.2 指定目的地
        FileOutputStream fos = new FileOutputStream( name: "D:\\demo22.txt");

        // 2.读写数据
        // 2.1 定义数组
        byte[] b=new byte[1024];
        // 2.2 定义长度
        int len;
        // 2.3 循环读取
        while ((len = fis.read(b))!=-1) {
```

```
C:\Users\雷神\.jdks\corretto-1.8.0_312\bin\java.exe ...

Process finished with exit code 0
```

Key Promoter X
Command Run missed 64 time(s)
'Ctrl+Shift+F10' (Disable alert for this shortcut)

Build completed successfully in 2 sec, 767 ms (moments ago)