**每日作业卷答案**

**JavaSE第10天**

传智播客.黑马程序员

# 关卡1

## 训练案例1

### 训练描述

创建两个文件对象，分别使用相对路径和绝对路径创建。

### 操作步骤答案

**public** **class** Test01\_01 {

**public** **static** **void** main(String[] args) {

// 创建文件对象：绝对路径

File f1 = **new** File("c:/aaa/a.txt");

// 创建文件对象：相对路径

File f2 = **new** File("c:/aaa/","a.txt");

}

}

## 训练案例2

### 训练描述

检查C盘下是否存在文件a.txt,如果不存在则创建该文件。

### 操作步骤答案

**public** **class** Test01\_02 {

**public** **static** **void** main(String[] args) **throws** IOException{

// 创建文件对象：绝对路径

File f = **new** File("c:/a.txt");

// 如果文件不存在，则创建文件

**if**(!f.exists()) {

f.createNewFile();

}

}

}

## 训练案例3

### 训练描述

在C盘下创建一个名为bbb的文件夹。

### 操作步骤答案

**public** **class** Test01\_03 {

**public** **static** **void** main(String[] args) {

// 创建文件对象

File f = **new** File("c:/bbb");

// 创建单级文件夹

f.mkdir();

}

}

## 训练案例4

### 训练描述

在C盘下创建一个名为ccc的文件夹，要求如下：

1.ccc文件夹中要求包含bbb子文件夹

2.bbb子文件夹要求包含aaa文件夹

### 操作步骤答案

**public** **class** Test01\_04 {

**public** **static** **void** main(String[] args) {

// 创建文件对象

File f = **new** File("c:/ccc/bbb/aaa");

// 创建多级文件夹

f.mkdirs();

}

}

## 训练案例5

### 训练描述

将C盘下a.txt文件删除

将C盘下aaa文件夹删除,要求文件夹aaa是一个空文件夹

### 操作步骤答案

**public** **class** Test01\_05 {

**public** **static** **void** main(String[] args) {

// 创建文件对象

File f = **new** File("c:/a.txt");

// 删除文件

f.delete();

// 创建文件夹对象

File dir = **new** File("c:/aaa");

// 删除文件夹

dir.delete();

}

}

## 训练案例6

### 训练描述

获取C盘aaa文件夹中b.txt文件的文件名，文件大小，文件的绝对路径和父路径等信息，并将信息输出在控制台。

### 操作步骤答案

**public** **class** Test01\_06 {

**public** **static** **void** main(String[] args) {

// 创建文件对象

File f = **new** File("c:/aaa/b.txt");

// 获得文件名

String filename = f.getName();

// 获得文件大小

**long** filesize = f.length();

// 获得文件的绝对路径

String path = f.getAbsolutePath();

// 获得父文件夹路径，返回字符串

String parentPath = f.getParent();

// 获得父文件夹路径，返回文件对象

File parentFile = f.getParentFile();

// 输出信息

System.***out***.println("文件名：" + filename);

System.***out***.println("文件大小：" + filesize);

System.***out***.println("文件路径：" + path);

System.***out***.println("文件父路径：" + parentPath);

System.***out***.println("文件父路径：" + parentFile);

}

}

## 训练案例7

### 训练描述

1.判断File对象是否文件,是文件则输出：xxx是一个文件，否则输出：xxx是不是一个文件。 2.判断File对象是否文件夹,是文件夹则输出：xxx是一个文件夹，否则输出：xxx是不是一个文件夹。(xxx是文件名或文件夹名)

### 操作步骤答案

**public** **class** Test01\_07 {

**public** **static** **void** main(String[] args) {

// 创建文件对象

File f1 = **new** File("c:/b.txt");

// 判断是否是一个文件

**if**(f1.isFile()) {

System.***out***.println(f1.getName()+"是一个文件");

} **else** {

System.***out***.println(f1.getName()+"不是一个文件");

}

// 创建文件对象

File f2 = **new** File("c:/aaaa");

// 判断是否是一个文件夹

**if**(f1.isDirectory()) {

System.***out***.println(f1.getName()+"是一个文件夹");

} **else** {

System.***out***.println(f1.getName()+"不是一个文件夹");

}

}

}

## 训练案例8

### 训练描述

获取指定文件夹下所有的文件，并将所有文件的名字输出到控制台。

注意：不包含子文件夹下的文件

### 操作步骤答案

**public** **class** Test01\_08 {

**public** **static** **void** main(String[] args) {

// 创建文件对象

File f = **new** File("c:/aaa");

// 获得文件夹下所有文件

File[] files = f.listFiles();

// 遍历文件数组

**for** (File file : files) {

// 将文件的名字打印到控制台

System.***out***.println(file.getName());

}

}

}

## 训练案例9

### 训练描述

求1到n的和(n>=100 && n<=200)

### 操作步骤答案

**public** **class** Test01\_09 {

**public** **static** **void** main(String[] args) {

**int** result = *sum*(100);

System.***out***.println(result);

}

// 递归求和方法

**public** **static** **int** sum(**int** num) {

// 递归出口

**if**(num == 1) **return** 1;

**return** num + *sum*(num - 1);

}

}

## 训练案例10

### 训练描述

求整数n的阶乘(n <=10)

### 操作步骤答案

**public** **class** Test01\_10 {

**public** **static** **void** main(String[] args) {

**int** result = *jc*(10);

System.***out***.println(result);

}

// 递归求阶乘方法

**public** **static** **int** jc(**int** num) {

// 递归出口

**if**(num == 1) **return** 1;

**return** num \* *jc*(num - 1);

}

}

# 关卡2

## 训练案例1

### 训练描述

键盘录入一个文件路径，根据文件路径创建File对象

如果输入的文件路径对应的文件不存在，则创建该文件。

如果输入的文件路径对应的文件已经存在了，则获得文件大小并输出。

### 操作步骤答案

**public** **class** Test02\_01 {

**public** **static** **void** main(String[] args) {

**try** {

// 创建键盘录入对象

Scanner sc = **new** Scanner(System.***in***);

// 提示用户输入一个文件路径

System.***out***.println("请输入一个文件路径:");

// 接收用户输入的文件路径

String filePath = sc.nextLine();

// 根据文件路径创建文件对象

File file = **new** File(filePath);

// 判断文件是否存在

**if**(!file.exists()) {

// 不存在，则创建文件

file.createNewFile();

} **else** {

// 输出文件大小

System.***out***.println("文件大小为：" + file.length());

}

} **catch** (IOException e) {

e.printStackTrace();

}

}

}

## 训练案例2

### 训练描述

键盘录入一个文件路径，根据文件路径创建文件对象，判断是文件还是文件夹

如果是文件，则输出文件的大小

如果是文件夹，则计算该文件夹下所有文件大小之和并输出(不包含子文件夹)。

### 操作步骤答案

**public** **class** Test02\_02 {

**public** **static** **void** main(String[] args) {

// 创建键盘录入对象

Scanner sc = **new** Scanner(System.***in***);

// 提示用户输入一个文件路径

System.***out***.println("请输入一个文件路径:");

// 接收用户输入的文件路径

String filePath = sc.nextLine();

// 根据文件路径创建文件对象

File file = **new** File(filePath);

// 判断是否是文件

**if**(file.isFile()) {

// 输出文件大小

System.***out***.println("文件大小为：" + file.length());

} **else** {

// 是文件夹

// 定义变量统计文件大小之和

**int** length = 0;

// 获得该文件夹下的所有文件

File[] files = file.listFiles();

// 遍历文件数组

**for** (File f : files) {

**if**(f.isFile()) {

// 累加文件大小

length += f.length();

}

}

// 输出文件夹文件的总大小

System.***out***.println("文件总大小为：" + length);

}

}

}

## 训练案例3

### 训练描述

递归遍历将指定文件夹的所有文件(包括所有子文件夹的文件)的全路径输出在控制台。

### 操作步骤答案

**public** **class** Test02\_03 {

**public** **static** **void** main(String[] args) {

// 创建文件夹对象

File dir = **new** File("c:/aaa");

// 调用递归方法

*printFiles*(dir);

}

/\*

\* 递归方法：遍历指定文件夹的所有文件

\*/

**public** **static** **void** printFiles(File dir){

// 获得该文件夹下的所有文件

File[] files = dir.listFiles();

// 遍历文件数组

**for** (File f : files) {

**if**(f.isFile()) { // 是文件

System.***out***.println(f.getAbsolutePath());

} **else** { // 是文件夹，递归调用当前方法

*printFiles*(f);

}

}

}

}

## 训练案例4

### 训练描述

获得指定文件夹下所有的java文件(不考虑子文件夹的)并输出到控制台

### 操作步骤答案

**public** **class** Test02\_04 {

**public** **static** **void** main(String[] args) {

// 创建文件夹对象

File file = **new** File("c:/aaa");

// 获得该文件夹下的所有java文件

File[] files = file.listFiles(**new** FileFilter() {

@Override

**public** **boolean** accept(File pathname) {

// 判断文件后缀名是否.java文件

**if**(pathname.getName().endsWith(".java")) **return** **true**;

**return** **false**;

}

});

// 遍历文件数组,打印所有的Java文件路径

**for** (File f : files) {

System.***out***.println(f.getAbsolutePath());

}

}

}

## 训练案例5

### 训练描述

键盘录入一个文件夹路径，删除该路径下的文件夹。

要求：录入的文件夹里面要有多个文件，不能包含有子文件夹。

提示：如果文件夹里面有文件，则需要先将文件删除才能删除文件夹。

### 操作步骤答案

**public** **class** Test02\_05 {

**public** **static** **void** main(String[] args) {

// 创建键盘录入对象

Scanner sc = **new** Scanner(System.***in***);

// 提示用户输入一个文件路径

System.***out***.println("请输入一个文件夹路径:");

// 接收用户输入的文件夹路径

String dirPath = sc.nextLine();

// 根据文件夹路径创建文件对象

File dir = **new** File(dirPath);

// 获得该文件夹下的所有文件

File[] files = dir.listFiles();

// 遍历文件数组删除每一个文件

**for** (File file : files) {

// 删除文件

file.delete();

}

// 删除文件夹

System.***out***.println(dir.delete()?"删除成功":"删除失败");

}

}

## 训练案例6

### 训练描述

从键盘接收一个文件夹路径,获得该文件夹大小并输出到控制台。

### 操作步骤答案

**public** **class** Test02\_06 {

**public** **static** **void** main(String[] args) {

// 创建键盘输入对象

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("请输入一个文件夹路径：");

// 接收用户输入的路径

String filePath = sc.nextLine();

// 根据路径字符串创建文件对象

File dir = **new** File(filePath);

// 调用calculate方法获得文件大小

**long** length = *calculate*(dir);

System.***out***.println("文件夹总大小：" + length);

}

/\*

\* 获得文件夹的大小

\*/

**public** **static** **long** calculate(File dir){

**long** length = 0;

// 获得文件数组

File files[] = dir.listFiles();

// 遍历数组

**for** (File file : files) {

// 判断是否是文件

**if**(file.isFile()) {

length += *calculate*(file);

} **else** {

length += file.length();

}

}

**return** length;

}

}

# 关卡3

## 训练案例1

### 训练描述

使用文件过滤器筛选将指定文件夹下的小于200K的小文件获取并打印(包括所有子文件夹的文件)。

### 操作步骤答案

**public** **class** Test03\_01 {

**public** **static** **void** main(String[] args) {

*filterFile*(**new** File("c:/aaa"));

}

/\*

\* 过滤器筛选将指定文件夹下的小于200K的小文件获取并打印

\*/

**public** **static** **void** filterFile(File f){

**if**(f.isFile()){

System.***out***.println(f + "不是文件夹");

**return**;

}

**if**(!f.exists()) **return**;

// 获得文件数组

File files[] = f.listFiles(**new** FileFilter() {

@Override

**public** **boolean** accept(File pathname) {

// 判断是否是文件夹，如果是则返回true

**if**(pathname.isDirectory()) **return** **true**;

// 判断是否是隐藏文件

**if**(pathname.isHidden()) **return** **false**;

// 获得文件大小

**long** length = pathname.length();

**if**(length / 1024 < 200) **return** **true**;

**return** **false**;

}

});

// 遍历数组

**for** (File file : files) {

**if**(file.isDirectory()) {

*filterFile*(file);

**continue**;

}

System.***out***.println(file);

}

}

}

## 训练案例2

### 训练描述

键盘录入一个文件夹路径，删除该路径下的文件夹。

要求：文件夹中包含有子文件夹

### 操作步骤答案

**public** **class** Test03\_02 {

**public** **static** **void** main(String[] args) {

// 获得文件夹对象

File dir = *inputFile*();

// 删除文件夹

*delete*(dir);

}

/\*

\* 根据用户输入的路径字符串创建文件对象

\*/

**private** **static** File inputFile() {

// 创建键盘输入对象

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("请输入一个文件夹路径：");

// 接收用户输入的路径

String filePath = sc.nextLine();

// 根据路径字符串创建文件对象

File dir = **new** File(filePath);

**if**(!dir.exists()) **throw** **new** RuntimeException("你输入的文件路径不存在！！！");

// 判断是否是文件夹

**if**(!dir.isDirectory()){

**throw** **new** RuntimeException("输入的路径不是文件夹路径！！！");

}

**return** dir;

}

/\*

\* 删除文件夹

\*/

**public** **static** **void** delete(File dir) {

// 获取所有文件

File[] files = dir.listFiles();

// 遍历子目录,删除子目录

**for** (File file : files) {

**if** (file.isFile()) {

**boolean** success = file.delete();

// 判断是否删除成功,主要用于测试自己写的代码

**if** (success) {

System.***out***.println(file + "删除成功");

} **else** {

System.***out***.println(file + "正在使用,删除失败");

}

} **else** {

// 如果是文件夹递归删除

*delete*(file);

}

}

// 来到这里说明,该文件夹为空了.

**boolean** success = dir.delete();

// 判断是否删除成功,主要用于测试自己写的代码

**if** (success) {

System.***out***.println(dir + "删除成功");

} **else** {

System.***out***.println(dir + "正在使用,删除失败");

}

}

}

## 训练案例3

### 训练描述

键盘录入一个文件夹路径,统计该文件夹(包含子文件夹)中每种类型的文件及个数

注意:用文件类型(后缀名,不包含.(点),如："java","txt")作为key,

用个数作为value,放入到map集合中,并用两种方式遍历map集合

例如：

doc 的类型的文件有 3 个

java 的类型的文件有 5 个

txt 的类型的文件有 7 个

### 操作步骤答案

**public** **class** Test03\_03 {

**public** **static** **void** main(String[] args) {

File dir = *inputFile*();

// 创建map集合

Map<String,Integer> map = **new** HashMap<String, Integer>();

*getFileType*(dir,map);

// 遍历map

**for**(Entry<String,Integer> entry:map.entrySet()) {

System.***out***.println(entry.getKey() + " 类型的文件有 " + entry.getValue() + " 个 ");

}

}

**public** **static** **void** getFileType(File f,Map<String,Integer> map) {

**if**(f.isDirectory()) {

File files[] = f.listFiles();

**for** (File file : files) {

*getFileType*(file,map);

}

} **else** {

// 获得文件名

String fileName = f.getName();

// 获得文件扩展名

String key = fileName.substring(fileName.lastIndexOf(".") + 1);

// 判断map是否包含key

**if**(map.containsKey(key)){

map.put(key, map.get(key) + 1);

} **else** {

map.put(key, 1);

}

}

}

/\*

\* 根据用户输入的路径字符串创建文件对象

\*/

**private** **static** File inputFile() {

// 创建键盘输入对象

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("请输入一个文件夹路径：");

// 接收用户输入的路径

String filePath = sc.nextLine();

// 根据路径字符串创建文件对象

File dir = **new** File(filePath);

**if**(!dir.exists()) **throw** **new** RuntimeException("你输入的文件路径不存在！！！");

// 判断是否是文件夹

**if**(!dir.isDirectory()){

**throw** **new** RuntimeException("输入的路径不是文件夹路径！！！");

}

**return** dir;

}

}

## 训练案例4

### 训练描述

从键盘接收一个文件夹路径,把文件夹中的所有文件以及文件夹的名字按层级打印

例如:

aaa是文件夹,里面有bbb.txt,ccc.txt,ddd.txt这些文件,有eee这样的文件夹,eee中有fff.txt和ggg.txt,打印出层级来

aaa

bbb.txt

ccc.txt

ddd.txt

eee

fff.txt

ggg.txt

### 操作步骤答案

**public** **class** Test03\_04 {

**public** **static** **void** main(String[] args) {

// 创建键盘输入对象

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("请输入一个文件夹路径：");

// 接收用户输入的路径

String filePath = sc.nextLine();

// 根据路径字符串创建文件对象

File dir = **new** File(filePath);

// 打印目录结构

*printDir*(dir,0);

}

/\*

\* 打印目录结构

\* level:表示目录的层次

\*/

**public** **static** **void** printDir(File f,**int** level){

**for** (**int** i = 0; i < level; i++) {

System.***out***.print("\t");

}

System.***out***.println(f.getName());

**if**(f.isDirectory()) {

File[] files = f.listFiles();

**for** (File file : files) {

*printDir*(file,level + 1);

}

}

}

}