Paths类

Path用于来表示文件路径和文件

1）首先是final类Paths的两个static方法，如何从一个路径字符串来构造Path对象：

```
Path path = Paths.get("C:/", "Xmp");
Path path2 = Paths.get("C:/Xmp");

URI u = URI.create("file:///C:/Xmp/dd");
Path p = Paths.get(u);
```

2）FileSystems构造：

```
Path path3 = FileSystems.getDefault().getPath("C:/",
"access.log");
```

3）File和Path之间的转换，File和URI之间的转换：

```
File file = new File("C:/my.ini");
Path p1 = file.toPath();
p1.toFile();
file.toURI();
```

Files类

1）创建目录和文件：

```
try {
    Files.createDirectories(Paths.get("C://TEST"));
    if(!Files.exists(Paths.get("C://TEST")))
    Files.createFile(Paths.get("C://TEST/test.txt"));
} catch (IOException e) {
        e.printStackTrace();
}
```

2）文件复制：

```
Files.copy(Paths.get("C://my.ini"), System.out);
Files.copy(Paths.get("C://my.ini"), Paths.get("C://my2.ini"),
StandardCopyOption.REPLACE_EXISTING);
Files.copy(System.in, Paths.get("C://my3.ini"),
StandardCopyOption.REPLACE_EXISTING);
```

3）遍历一个目录和文件夹上面已经介绍了

单个目录

```
Path dir = Paths.get("D:\\webworkspace");
try(DirectoryStream<Path> stream =
Files.newDirectoryStream(dir)){
```

```java
    for(Path e : stream){
        System.out.println(e.getFileName());
    }
}catch(IOException e){

}
```

整个目录

```java
public static void main(String[] args) throws IOException{
    Path startingDir = Paths.get("C:\\apache-tomcat-8.0.21");
    List<Path> result = new LinkedList<Path>();
    Files.walkFileTree(startingDir, new
FindJavaVisitor(result));
    System.out.println("result.size()=" + result.size());
}

private static class FindJavaVisitor extends
SimpleFileVisitor<Path>{
    private List<Path> result;
     public FindJavaVisitor(List<Path> result){
         this.result = result;
     }
     @Override
     public FileVisitResult visitFile(Path file,
BasicFileAttributes attrs){
         if(file.toString().endsWith(".java")){
             result.add(file.getFileName());
         }
         return FileVisitResult.CONTINUE;
     }
}
```

4）读取文件属性：

```java
Path zip = Paths.get(uri);
System.out.println(Files.getLastModifiedTime(zip));
System.out.println(Files.size(zip));
System.out.println(Files.isSymbolicLink(zip));
System.out.println(Files.isDirectory(zip));
System.out.println(Files.readAttributes(zip, "*"));
```

5）读取和设置文件权限：

```java
Path profile = Paths.get("/home/digdeep/.profile");
PosixFileAttributes attrs = Files.readAttributes(profile,
PosixFileAttributes.class);// 读取文件的权限
Set<PosixFilePermission> posixPermissions =
attrs.permissions();
posixPermissions.clear();
String owner = attrs.owner().getName();
String perms =
PosixFilePermissions.toString(posixPermissions);
System.out.format("%s %s%n", owner, perms);

posixPermissions.add(PosixFilePermission.OWNER_READ);
posixPermissions.add(PosixFilePermission.GROUP_READ);
posixPermissions.add(PosixFilePermission.OTHERS_READ);
posixPermissions.add(PosixFilePermission.OWNER_WRITE);

Files.setPosixFilePermissions(profile, posixPermissions); // 设置文件的权限
```