第一题

使用绝对路径，在D盘创建一个testIO文件夹，然后再testIO文件中创建一个1.txt文件中；使用相对路径，在当前项目下创建一个testIO文件夹，然后再testIO文件中创建一个2.txt文件中

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

File file = new File("testIO/2.txt");

// File file1 = new File("D:/testIO/1.txt");

File parent = file.getParentFile();

System.out.println(parent);

if (!parent.exists())

parent.mkdirs();

boolean res = parent.exists();

if (res && !file.exists()) {

file.createNewFile();

}

}

第二题

文件夹的剪切

public static void main(String[] args) {

File f1 = new File("output/");

File f2 = new File("test/");

try {

cut(f1, f2);

System.out.println("剪切完成");

} catch (Exception e) {

e.printStackTrace();

}

}

public static void cut(File file, File target) throws Exception {

if (file.exists()) {

if (!target.exists())

target.mkdirs();

if (file.isFile()) {

File scr = new File(target, file.getName());

copy(file, scr);

} else if (file.isDirectory()) {

File destFile = new File(target, file.getName());

destFile.mkdirs();

File[] children = file.listFiles();

for (File temp : children) {

cut(temp, destFile);

}

}

file.delete();

}

}

public static void copy(File file, File target) throws Exception {

FileInputStream fis = new FileInputStream(file);

FileOutputStream fos = new FileOutputStream(target);

int len = 0;

byte[] buffer = new byte[1024];

while ((len = fis.read(buffer)) > 0) {

fos.write(buffer);

}

fos.close();

fis.close();

}

第三题

删除当前项目文件夹

public static void main(String[] args) {

File file = new File("ttt/");

delete(file);

}

public static void delete(File file) {

if (file.exists()) {

if (file.isFile()) {

file.delete();

} else if (file.isDirectory()) {

File[] children = file.listFiles();

for (File temp : children) {

delete(temp);

}

}

file.delete();

}

}

第四题

文件夹的复制

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

copy("testIO/", "ttt/");

}

public static void copy(String source, String target) throws Exception {

File file = new File(target);

if (!file.exists())

file.mkdirs();

File[] files = new File(source).listFiles(); // 获取data目录下的所有子文件和目录

for (File tmp : files) {

if (tmp.isFile()) {

InputStream fis = new FileInputStream(tmp);

String fileName = tmp.getName();

OutputStream fos = new FileOutputStream(target + fileName);

byte[] buffer = new byte[8192];

int len = 0;

while ((len = fis.read(buffer)) > 0) {

fos.write(buffer, 0, len);

}

fos.close();

fis.close();

} else {

String ss = tmp.getName();

File stmp = new File(target + "/" + ss + "/");

if (!stmp.exists()) {

stmp.mkdirs();

}

copy(tmp.getAbsolutePath() + "/", stmp.getAbsolutePath() + "/");

}

}

}

第五题

1.声明一个Message类，包括：发送者、接收者、消息内容、发送时间

2.创建一个Message对象，并写到message.data文件中，再次读取显示

public class Messagae implements Serializable {

private String input;

private String output;

private String message;

private String data;

@Override

public String toString() {

return "Messagae [input=" + input + ", output=" + output + ", message=" + message + ", data=" + data + "]";

}

public String getInput() {

return input;

}

public void setInput(String input) {

this.input = input;

}

public String getOutput() {

return output;

}

public void setOutput(String output) {

this.output = output;

}

public String getMessage() {

return message;

}

public void setMessage(String message) {

this.message = message;

}

public String getData() {

return data;

}

public void setData(String data) {

this.data = data;

}

}

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

Messagae ss = new Messagae();

ss.setInput("123");

ss.setOutput("1234");

ss.setMessage("2344");

DateFormat dd = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

Date now = new Date();

String date = dd.format(now);

ss.setData(date);

ss.toString();

BufferedWriter bw = new BufferedWriter(new FileWriter("message.data"));

String s = ss.toString();

while (ss != null) {

bw.write(s);

}

bw.close();

}

第六题

把一篇文档字符编码为GBK，复制到当先项目下，字符编码为UTF-8

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

InputStream rr = new FileInputStream("data/data1.txt");

Reader r = new InputStreamReader(rr, "utf-8");

OutputStream w = new FileOutputStream("data/data3.txt");

int len = 0;

while ((len = r.read()) != -1) {

System.out.print((char) len);

w.write((char) len);

}

w.close();

r.close();

rr.close();

}