实验一：体验异常的捕捉

实验描述：

创建一个类，在这个类中定义可能发生异常的一些方法，并在这些方法中捕捉异常，合理处理。在主方法中执行这些方法。

实验要求：

1. 静态方法 method1()

   提示由键盘输入两个整数，并计算两个整数的商。需要考虑输入数据类型不匹配的异常和除数为 0 的异常。如果发生异常，提示重新输入数据直到数据和商合法为止。最后输出输入的两个数及二者的商。

2. 静态方法 method2()

   创建一个合适长度的数组，数组中元素的值随机赋值。赋值后提示用户键盘输入要查看的数组元素的下标，考虑数组下标越界的异常。如果发生异常，同样给出具体提示，并要求重新输入下标直到下标合法并打印出此元素的值。

3. 在主方法里调用这两个方法进行测试

实验过程

核心代码展示：

```
//method1
public static void method1() {
        Scanner scanner = new Scanner(System.in);
        int num1=0, num2=0;
        int result = 0;
        boolean validInput = false;

        while (!validInput) {
            try {
                System.out.println("请输入两个整数:");
                num1 = scanner.nextInt();
                num2 = scanner.nextInt();

                result = num1 / num2;
                validInput = true;
            } catch (InputMismatchException e) {
                System.out.println("输入的数据类型不匹配，请重新输入整数。");
                scanner.nextLine(); // 清空输入缓冲区
            } catch (ArithmeticException e) {
                System.out.println("除数不能为0，请重新输入。");
                scanner.nextLine(); // 清空输入缓冲区
            }
        }
        System.out.print("num1 is "+num1);
        System.out.print(", num2 is "+num2);
        System.out.println(", result of num1/num2 is: "+result);
    }
```

```java
//method2
public static void method2() {
        Scanner scanner = new Scanner(System.in);

        System.out.print("请输入数组长度: ");
        int length = scanner.nextInt();
        int[] array = new int[length];

        // 随机赋值
        for (int i = 0; i < length; i++) {
            array[i] = (int) (Math.random() * 100);
        }

        int index = -1;
        boolean validIndex = false;

        while (!validIndex) {
            try {
                System.out.print("请输入要查看的数组元素的下标: ");
                index = scanner.nextInt();

                if (index < 0 || index >= length) {
                    System.out.println("下标越界，请重新输入。");
                } else {
                    validIndex = true;
                }
            } catch (InputMismatchException e) {
                System.out.println("输入的数据类型不匹配，请重新输入整数。");
                scanner.nextLine(); // 清空输入缓冲区
            }
        }
```

涉及知识点：

try-catch语句：

- 捕捉多个异常

```java
try {
    // 一些可能抛出异常的代码
    int[] arr = {1, 2, 3};
    System.out.println(arr[5]);
} catch (ArrayIndexOutOfBoundsException e) {
    // 处理ArrayIndexOutOfBoundsException异常
    System.out.println("数组索引越界！");
} catch (NullPointerException e) {
    // 处理NullPointerException异常
    System.out.println("空指针异常！");
}
```

- 使用finally语句：

```java
FileWriter writer = null;
```

```
try {
    // 一些可能抛出异常的代码
    writer = new FileWriter("file.txt");
    writer.write("Hello, world!");
} catch (IOException e) {
    // 处理IOException异常
    System.out.println("文件写入错误! ");
} finally {
    // 无论是否发生异常，都会执行finally块中的代码
    try {
        if (writer != null) {
            writer.close();
        }
    } catch (IOException e) {
        // 处理关闭文件时可能发生的异常
        System.out.println("关闭文件错误! ");
    }
}
```

实验结果

```
请输入两个整数：
5
sa
输入的数据类型不匹配，请重新输入整数。
请输入两个整数：
5
0
除数不能为0，请重新输入。
请输入两个整数：
5
3
num1 is 5, num2 is 3, result of num1/num2 is: 1
请输入数组长度：15
请输入要查看的数组元素的下标：-1
下标越界，请重新输入。
请输入要查看的数组元素的下标：16
下标越界，请重新输入。
请输入要查看的数组元素的下标：8
选择的下标：8
对应的数组元素值：58
```

实验二：自定义异常

实验描述：

一个用户类，用户类的成员变量有用户姓名、用户年龄、用户工资以及用户邮箱。考虑对自定义异常类，对设定的用户工资和设定的邮箱进行输入数据的合法性检查。

实验要求：

1. 创建一个 MyException 异常类，继承自 RuntimeException。是本项目的异常类的超类
2. 创建 MalformedSalary 类和 MalformedEmailAddress 类，继承自 MyException 类。MalformedSalary 类用于对输入 salary 的合法性检查；MalformedEmailAddress 用于对输入 email address 的合法性检查
3. 一个 User 类
   (1) name, age, salary, emailAddress 四个成员变量
   (2) 无参构造方法将成员变量设置为默认值
   (3) 各成员变量的 getter 和 setter 方法
   (4) setSalary()方法中要进行对设置的 salary 数据类型做检查(判断符合你所定义的 salary 数据类型，不符合的话抛出异常)
   (5) setEmailAddress()方法中要对输入的 emial address 做合法性的检查。如果输入的 email address 不合法，提供再输入一次的机会，对新输入的 email address 同样要做合法性检查
   (6) toString()方法返回 User 的信息
4. UserTester 类
   创建 User 的对象，并调用相应方法进行测试

实验过程

核心代码：

```
//MalformSalary类
//对父类中的detailMessage方法重写
    public String detailMessage(){
        return "salary格式错误";
    }
    //使用Integer.parseInt判断是否合法
    public boolean judge(){
        int num=0;
            try {
                num = Integer.parseInt(salary);
                if (num < 0){
                    System.out.println(detailMessage());
                    return false;
                }
            } catch (NumberFormatException e) {
                System.out.println(detailMessage());
                return false;
            }
            return true;

    }
```

```
//MalformedEmailAddress类
//对父类中的detailMessage方法重写
```

```
public String detailMessage(){
        return "EmailAdress格式不符合";
    }
//判断是否符合email格式
    public boolean judge(){
        String tag1="@qq.com";
        String tag2=" ";
        int index=emailAddress.indexOf(tag1);//必须含有tag1关键字
        boolean contains2=emailAddress.contains(tag2);
        //包含"@qq.com"且不包含空格
        if((index!=emailAddress.length()-tag1.length())||contains2){
            System.out.println(detailMessage());
            return false;
        }else{
            //"@qq.com"前只可有数字or字母
            for(int i=0;i<index;i++){
                    char c=emailAddress.charAt(i);
                    if(!(Character.isDigit(c)||Character.isLetter(c))){
                        System.out.println(detailMessage());
                        return false;
                    }
            }
        }
        return true;
    }
```

实验结果

```
User name Alex
Age=20
Salary=1000
EmailAddress='12345@qq.com

salary格式错误


EmailAdress格式不符合
请重新输入
deng123@qq.com
User name Jared
Age=22
Salary=NaN
EmailAddress='deng123@qq.com
```

实验三：Java版的grep

实验描述：

请实现一个 Java 版的 grep，grep 是一个常用的文本搜索工具，它可以在一个或多个文件中查找特定的字符串，然后输出包含该字符串的行。本次题目要求在当前目录的文本文件（可以指定文件名）中搜索特定的字符串，然后把符合要求的行以及行号输出到 output.txt。

实验要求：

1. 从命令行参数中获取需要查找的字符串和需要查找的文件名（如果没有指定文件名，则在当前目录下的所有文本文件中查找）。
2. 如果查找到符合要求的行，则将该行以及行号输出到 output.txt 文件中。
3. 如果没有找到符合要求的行，则输出“未找到符合要求的行”。
4. 要求程序具有良好的代码风格和注释，尤其是需要清晰明了地注释代码中的关键步骤。
5. 程序应当对 txt 文件进行逐行读取，并在每次成功匹配时立即写入到 output.txt 中，而不是等全部的 txt 文件读取完。
6. 合理使用 File，PrintWriter 等类来实现上述功能。
7. 对异常情况进行处理，采用或设计合适的异常。

实验过程

核心思路

首先获取当前文件路径，将txt文件存入数组。遍历数组中的每个txt文件的每一行，判断是否含有所查找的字符串searchString，若有，将当前文件名称即searchString出现行数存入output.txt文件中

核心代码

```
//遍历当前目录下所有的txt文件，并遍历所有txt文件，调用searchFile来完成具体操作
try {
            File[] files;
            if (fileName != null) {
                File file = new File(fileName);
                files = new File[]{file};
            } else {

                //创建currentDir来表示当前目录
                File currentDir = new File(".");
                //查看当前目录绝对路径的字符串表达
                System.out.println(currentDir.getAbsolutePath());
                //用ListFiles筛选出所有以".txt"结尾的文件，存入file数组
                files = currentDir.listFiles((dir, name) ->
name.toLowerCase().endsWith(".txt"));

            }
            //若在文件夹中没有找到符合要求的行
            if (files == null || files.length == 0) {
                System.out.println("未找到符合要求的行");
                return;
            }
            //创建PrintWriter对象，用于向output文件输入
            PrintWriter writer = new PrintWriter("output.txt");
            for (File file : files) {
                searchFile(file, searchString, writer);
            }
            writer.close();

            System.out.println("搜索完成，结果已保存到output.txt");
        } catch (IOException e) {
            System.out.println("发生IO异常：" + e.getMessage());
        }
```

```
//判断当前文件的所有行中是否含有searchString，若有，写入writer指代的文件中
private static void searchFile(File file, String searchString, PrintWriter
writer) throws IOException {
        //当前行号
        int lineNumber = 0;
        try (BufferedReader reader = new BufferedReader(new FileReader(file))) {
            //每行文本
            String line;
            //直至文件末尾
            while ((line = reader.readLine()) != null) {
                lineNumber++;
                if (line.contains(searchString)) {
                    //每次匹配成功立即写入output.txt
                    writer.println("文件：" + file.getName() + "，行号：" +
lineNumber + "，内容：" + line);
                }
            }
        }
```

```
    }
```

实验结果

1.txt

```
hello




omygod


test
```

2.txt

```
hello!


hi
```

output.txt

```
文件：1.txt，行号：1，内容：hello
文件：2.txt，行号：9，内容：hello!
```

实验四：网站外链管理系统

实验描述：
网站内容管理工作需要了解一个网站里有哪些外链，即指向非本站的链接。本题涉及的网站为学院网站，其域名为"xmu.edu.cn"。现需设计一个基于爬虫的网站外链管理系统来解决这个问题，它能找出学院网站里的每个页面的外链，并分门别类。

任务：
请分析问题并设计相应的对象，如网站、页面、链接、爬虫、爬虫调度等等。并使用 Java 语言实现一个简单的外链管理系统。

实验要求：
1. 设计并实现以下类：
   - Website：代表一个网站，包含网站的基本信息，如域名、首页 URL 等。
   - WebPage：代表一个网页，包含 URL、页面内容、外链等信息。
   - Link：代表一个链接，包含 URL、链接类型（本校、非本校）等信息。
   - Crawler：爬虫类，负责抓取指定网页的内容，并解析出外链。
   - CrawlerScheduler：爬虫调度类，负责调度并管理爬虫任务。

2. 实现一个简单的爬虫，可以从指定的网站开始抓取页面，并解析出其中链接。将链接按照本校和非本校分为内链和外链。对于内链，继续进行爬取，直到爬取完整个网站。

3. 爬虫调度类应当能够支持多线程爬取，以提高抓取速度。并且能合理安排爬虫任务，避免重复抓取同一网页。

4. 提供一个简单的用户界面，对网站(**首页为：https://informatics.xmu.edu.cn/**)进行爬取，并展示外链分类结果。

提示：
1. 可以考虑使用 [Jsoup](https://jsoup.org/) 库来解析 HTML 页面，提取外链。（从 IDEA 导入 jsoup 的一种方法是：Poject Structure-Project Settings-Libraries-From Maven…，搜索 jsoup，选择 org.jsoup 即可）
2. 使用 java.net.URL 类处理 URL 相关操作，如判断链接是否属于本校等。

评分标准：
1. 代码结构清晰，模块划分合理，符合面向对象设计原则。
2. 爬虫能够正确抓取网页并解析出外链。
3. 系统能够正确区分本校和非本校链接。
4. 用户界面简洁、易用，能够展示外链分类结果。
5. 用户界面应当展示出学校网站每个页面的外链，或者展示出部分重要网页的外链。如果能给出统计信息会更好。
5. CrawlerScheduler 类能够支持多线程抓取，通过启动多个爬虫线程来提高爬取速度。
6. CrawlerScheduler 类能合理安排爬虫任务，避免重复抓取同一网页

实验过程

- 区分本校和非本校

```
            //获取<a>标签元素的"href"属性的值
            String href = link.attr("abs:href");
```

```
            //System.out.println(href);
            //若为内链
            if(href.startsWith(root))
{//root="https://informatics.xmu.edu.cn/"
                //跳过下载的文件
                if (!(href.contains(".doc") || href.contains(".xls") ||
href.contains(".pdf")||href.contains(".rar")))
                    //加入集合
                    internalLinks.add(href);
            }
            else {
                //且不是校内网站
                if(!href.contains("xmu"))
                    outernalLinks.add(href);
            }
```

- 展示外链分类结果(存入csv)

```
            String csvFilePath2 = "outer.csv";
            try (FileWriter writer = new FileWriter(csvFilePath2)) {
                for (String item : outernalLinks) {
                    writer.append(item);
                    writer.append("\n");
                }
                writer.flush();
                System.out.println("outernalLinks has been written to the
CSV file.");
            } catch (IOException e) {
                e.printStackTrace();
            }
```

- 不重复爬取

  用flagLinks集合来保证，每次爬取前需确认链接不包含在flagLinks中，爬取后将其加入flagLinks

```
            for (String element : internalLinks) {
                if (!flagLinks.contains(element)) {
                    flagLinks.add(element);
                    System.out.println("从首页的:"+element+"开始爬取");

 CrawlerScheduler.getNextURL("https://informatics.xmu.edu.cn/", element,
flagLinks, outernalLinks);
                }
            }
```

- 爬取https://informatics.xmu.edu.cn/的所有内链

```
public static void getNextURL(String root,String url,Set<String>
flag,Set<String> outer){

        Crawler temp=new Crawler(root,url);
```

```
        if(temp.setDoc()) {

            Set<String> in = new HashSet<>();
            Set<String> out = new HashSet<>();
            temp.setLinks();
            in = temp.getInternalLinks();
            out = temp.getOuternalLinks();
            outer.addAll(out);

            Set<String> xorSet = new HashSet<>();
            xorSet.addAll(in);
            xorSet.removeAll(flag);
            for (String inter : in) {
                //若未爬取过,且同双胞胎链接未被爬取过
                if (!
(flag.contains(inter)||flag.contains(inter.substring(0,inter.length()-1))))
{
                    flag.add(inter);
                    System.out.println(inter);
                    getNextURL(root, inter, flag, outer);
                } else
                    continue;
            }
        }else
            return;

    }
```

- 获取网站域名

```
XMU.setDomain(bug.getDomain());//获取网站域名
            System.out.println("网站" + XMU.getLink().getUrl() + "的域名是：" +
XMU.getDomain());
```

实验结果

- 获取网站域名

```
E:\Java\jdk-17\bin\java.exe "-javaagent:D:\Program Files\JetBrains\I
域名为：informatics.xmu.edu.cn
网站https://informatics.xmu.edu.cn/的域名是：informatics.xmu.edu.cn
从首页的:https://informatics.xmu.edu.cn/info/1054/28439.htm开始爬取
```

- 获取所有https://informatics.xmu.edu.cn/的内链并存为inter.csv

  共获取了137条不含下载文件(rar、doc、xls、pdf)的、不含双胞胎链接的内链

132	https://informatics.xmu.edu.cn/faculty/cogsci/professor.htm
133	https://informatics.xmu.edu.cn/faculty/se/associate_prof.htr
134	https://informatics.xmu.edu.cn/research/platforms.htm
135	https://informatics.xmu.edu.cn/faculty/eietc/gcs.htm
136	https://informatics.xmu.edu.cn/info/1335/26389.htm
137	https://informatics.xmu.edu.cn/introduction/lxwm.htm
138	

- 获取所有https://informatics.xmu.edu.cn/的外链并存为outer.csv

  共获取了1701条不重复的外链

1679	http://www.fjedu.gov.cn/html/jyzt/ffdxwlxxwffz/spjq/2016/01/14/0db38427-122f-4722-b206-d0ecb8a7063e.html
1680	http://career.huawei.com/campus/
1681	mailto:epgis_recruit@sgcc.com.cn
1682	http://www.temple.edu/registrar/documents/calendars/
1683	mailto:icchina@microsoft.com
1684	mailto:wenzhenhua@gxlu.com.cn
1685	http://www.kingtopinfo.com/index.asp
1686	http://tech.ifeng.com/c/7zmvtkjL9ss
1687	http://www.seavo.com/PageView2.aspx?menuid=6&popid1=88&tid=30&bid=1395
1688	http://www.91boshuo.com/
1689	mailto:鏊戦€佽噩rjxyshsj@126.com
1690	mailto:jmatlin@temple.edu
1691	http://www.dataexa.com/school-recruitment
1692	http://www.dmc.isu.edu.tw/
1693	http://www.fjedu.gov.cn/html/jyzt/ffdxwlxxwffz/spjq/2016/01/14/1c5446ee-7836-49c9-aabe-e4ff3a5c521a.html
1694	https://baike.baidu.com/item/%E5%9B%BD%E5%AE%B6%E6%95%B0%E7%A0%81%E5%96%B7%E5%8D%B0%E5%B7%A5%E7%A8%8B%E
1695	http://www.renren.com/kczx
1696	mailto:angel.an@ehealth-china.com
1697	mailto:rbjobs@cmbchina.com
1698	http://www.fjedu.gov.cn/html/ffdxwlxxwffz/
1699	mailto:hr06@optosky.com
1700	mailto:zhangyong@corp.chinahr.com
1701	http://www.300188.cn%20鈥旀€?
1702	