《面向对象程序设计与实践》实验报告

实验名称： 《面向对象程序设计与实践》实验八

学 期： 2022春季学期 任课教师： 陆歌皓 学院 软件学院

专业： 计算机 学号： 20211120138 姓名： 薛凡豪 成绩：

# 一、实验源代码及运行结果：

## 1.检索简历

## （1）问题分析：

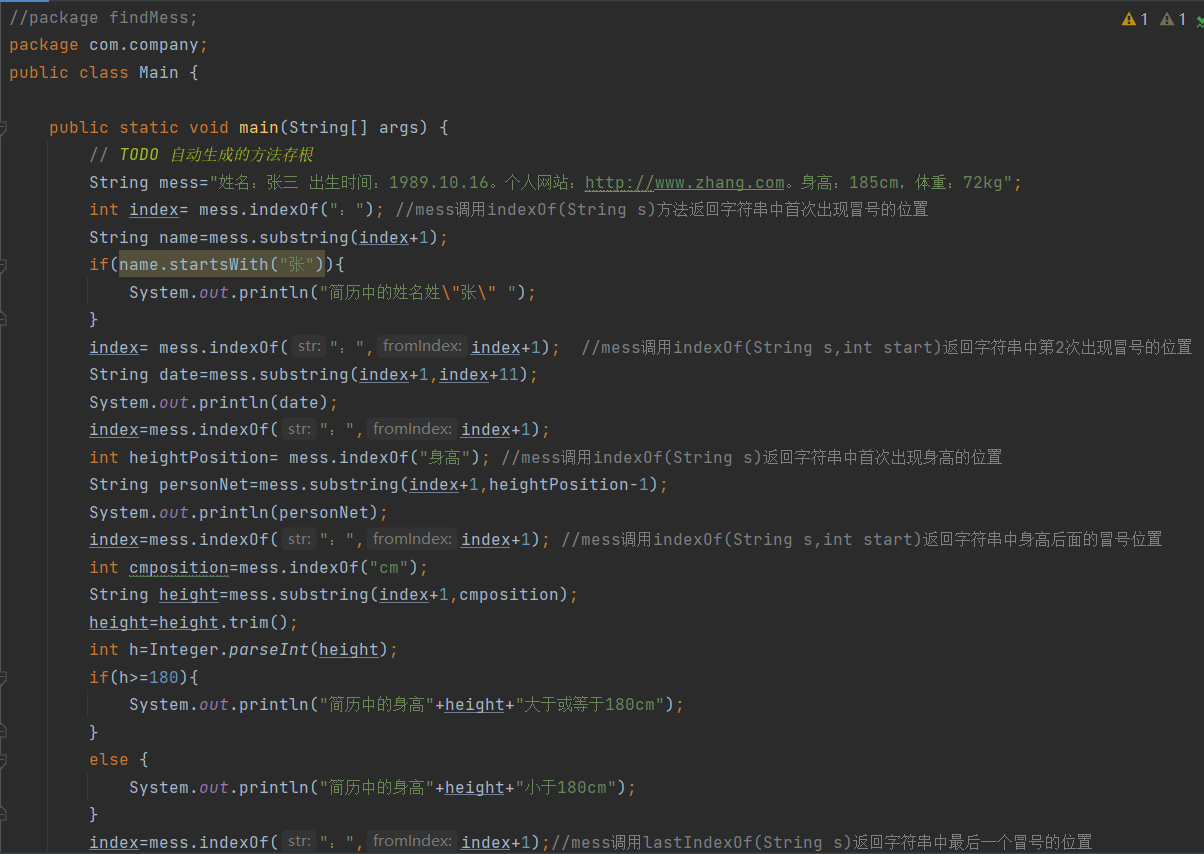
1.本实验主要需要运用的是字符串的使用（Srting）

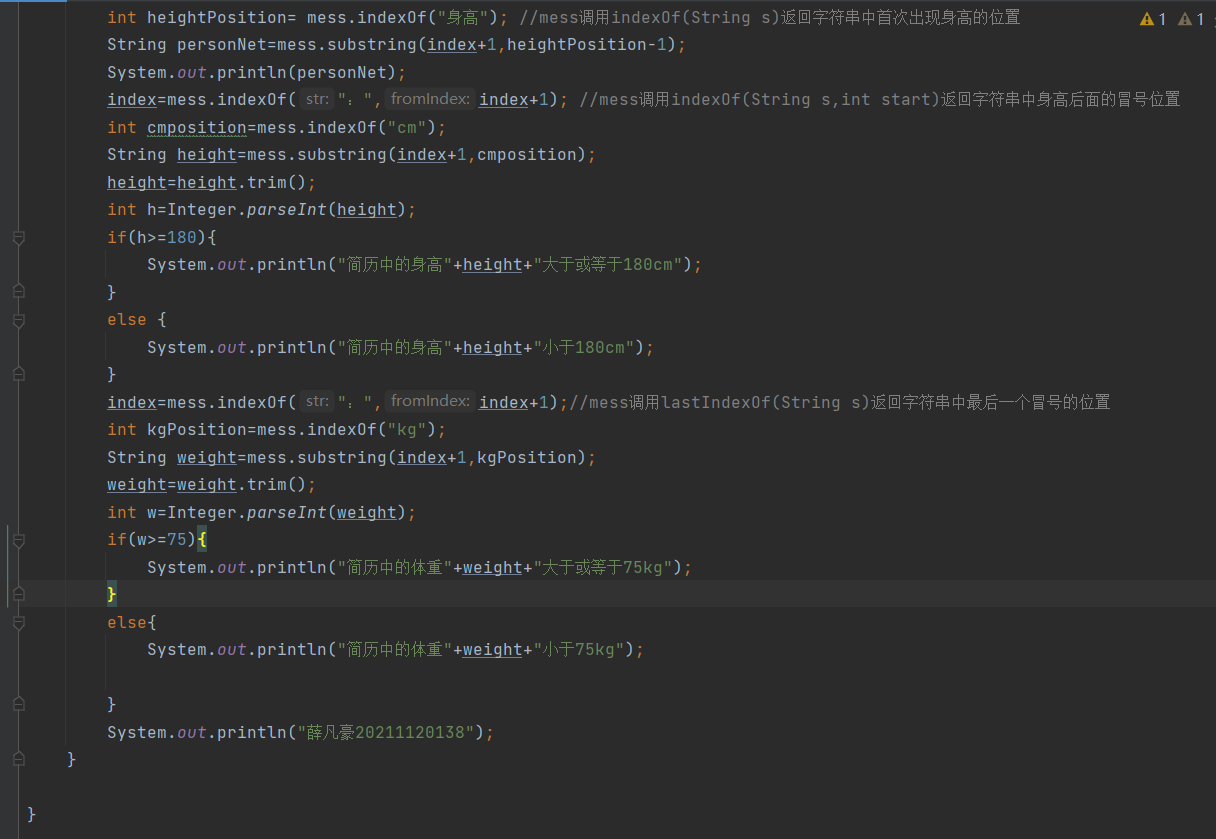
2.重点是掌握String类的常用方法和使用。

3.编写程序，判断简历中的姓名是否姓张，并且输出简历中的出生日期和个人网站，并判断身高是否大于180cm’，体重是否小于75kg

4.本实验用到了indexOf(int n)、substring(int index)等方法。

### （2）程序的运行和结果

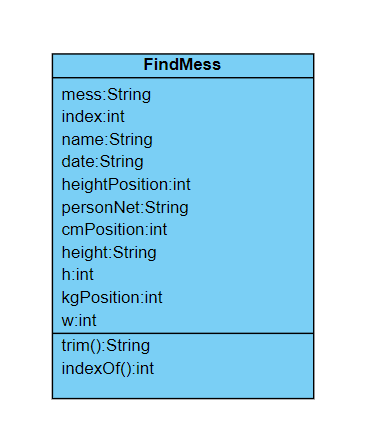




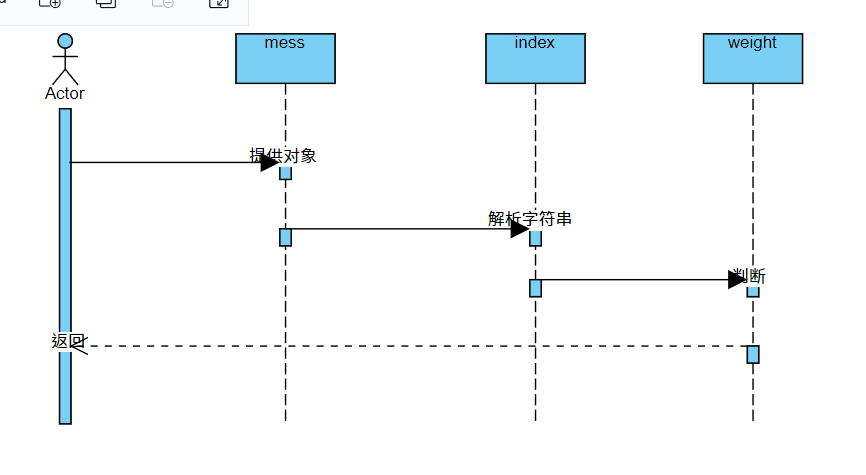


### （3）UML图

类图：



时序图：



### （4）源代码：

**Main.java**

//package findMess;  
package com.company;  
public class Main {  
  
    public static void main(String[] args) {  
        // TODO 自动生成的方法存根  
        String mess="姓名：张三 出生时间：1989.10.16。个人网站：http://www.zhang.com。身高：185cm，体重：72kg";  
        int index= mess.indexOf("："); //mess调用indexOf(String s)方法返回字符串中首次出现冒号的位置  
        String name=mess.substring(index+1);  
        if(name.startsWith("张")){  
            System.out.println("简历中的姓名姓\"张\" ");  
        }  
        index= mess.indexOf("：",index+1);  //mess调用indexOf(String s,int start)返回字符串中第2次出现冒号的位置  
        String date=mess.substring(index+1,index+11);  
        System.out.println(date);  
        index=mess.indexOf("：",index+1);  
        int heightPosition= mess.indexOf("身高"); //mess调用indexOf(String s)返回字符串中首次出现身高的位置  
        String personNet=mess.substring(index+1,heightPosition-1);  
        System.out.println(personNet);  
        index=mess.indexOf("：",index+1); //mess调用indexOf(String s,int start)返回字符串中身高后面的冒号位置  
        int cmposition=mess.indexOf("cm");  
        String height=mess.substring(index+1,cmposition);  
        height=height.trim();  
        int h=Integer.parseInt(height);  
        if(h>=180){  
            System.out.println("简历中的身高"+height+"大于或等于180cm");  
        }  
        else {  
            System.out.println("简历中的身高"+height+"小于180cm");  
        }  
        index=mess.indexOf("：",index+1);//mess调用lastIndexOf(String s)返回字符串中最后一个冒号的位置  
        int kgPosition=mess.indexOf("kg");  
        String weight=mess.substring(index+1,kgPosition);  
        weight=weight.trim();  
        int w=Integer.parseInt(weight);  
        if(w>=75){  
            System.out.println("简历中的体重"+weight+"大于或等于75kg");  
        }  
        else{  
            System.out.println("简历中的体重"+weight+"小于75kg");  
  
        }  
        System.out.println("薛凡豪20211120138");  
    }  
  
}    }

## 2.菜单的价格

### （1）问题分析：

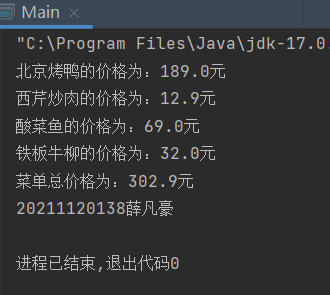
1.本次实验主要练习Scanner类的对象从字符串中解析数据

2.问题：编写一个程序，输出菜单中的价格数据，并计算出菜单的总价格。

3.实验运用了Scanner调用useDelimiter(String regex)将正则表达式作为分割标记，即与正则表达式regex匹配的字符串作为分割标记。

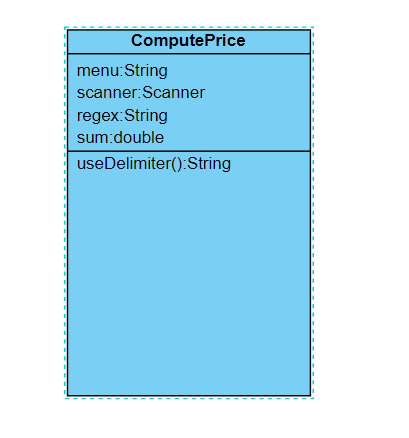
### （2）程序的运行和结果



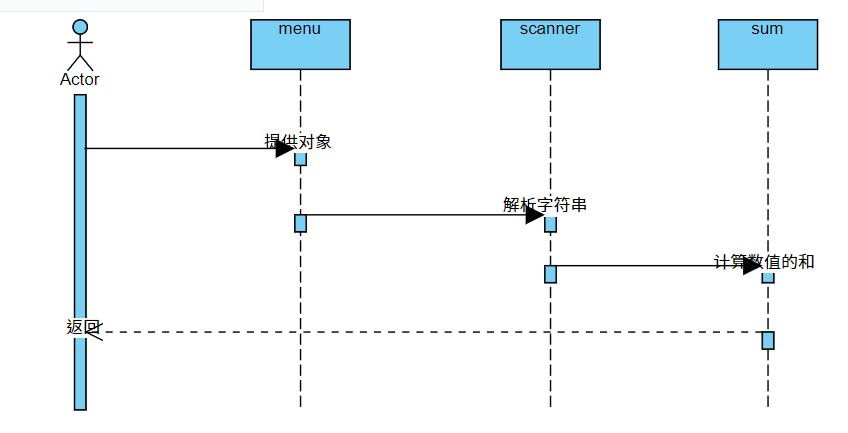


### （3）UML图

类图：



时序图：



### （4）源代码

Main.java

package com.company;  
  
import java.util.Scanner;  
public class Main {  
    public static void main(String args[]){  
        String s="北京烤鸭：189元，西芹炒肉：12.9元，酸菜鱼：69元，铁板牛柳：32元。";  
        s=s.replace("：", " ");  
        s=s.replace("，", " ");  
        s=s.replace("元", "");  
        s=s.replace("。", "");  
        double price,sum=0.0;  
        String name;  
        Scanner reader=new Scanner(s);  
        while(reader.hasNext()){  
            name=reader.next();  
            price=reader.nextDouble();  
            System.out.println(name+"的价格为："+price+"元");  
            sum+=price;  
        }  
        System.out.println("菜单总价格为："+sum+"元");  
        System.out.println("20211120138薛凡豪");  
    }  
}        }

## 3.比较日期

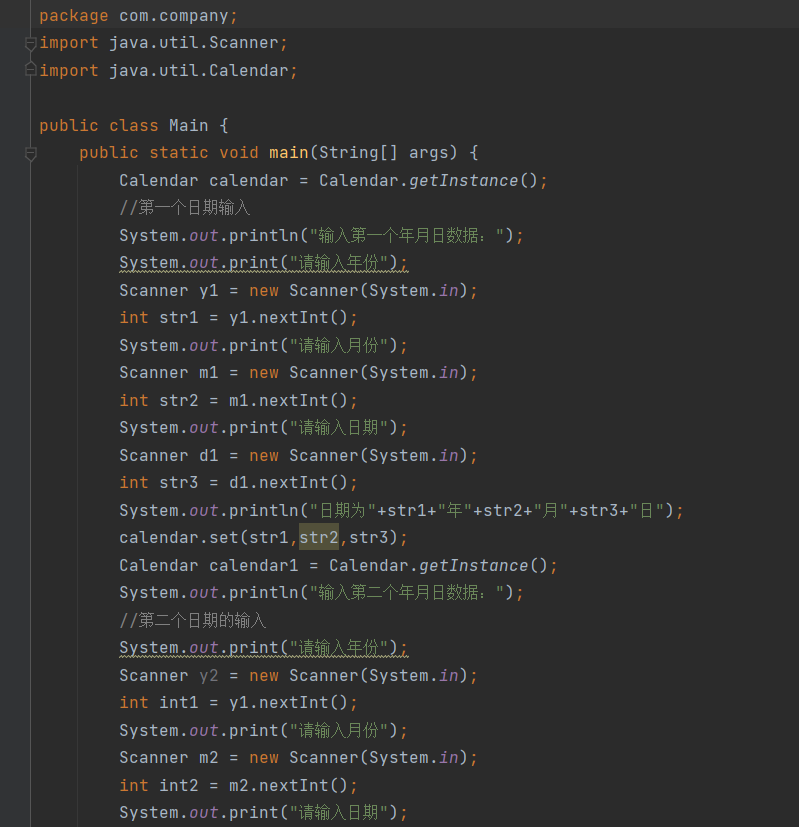
### (1)问题分析：

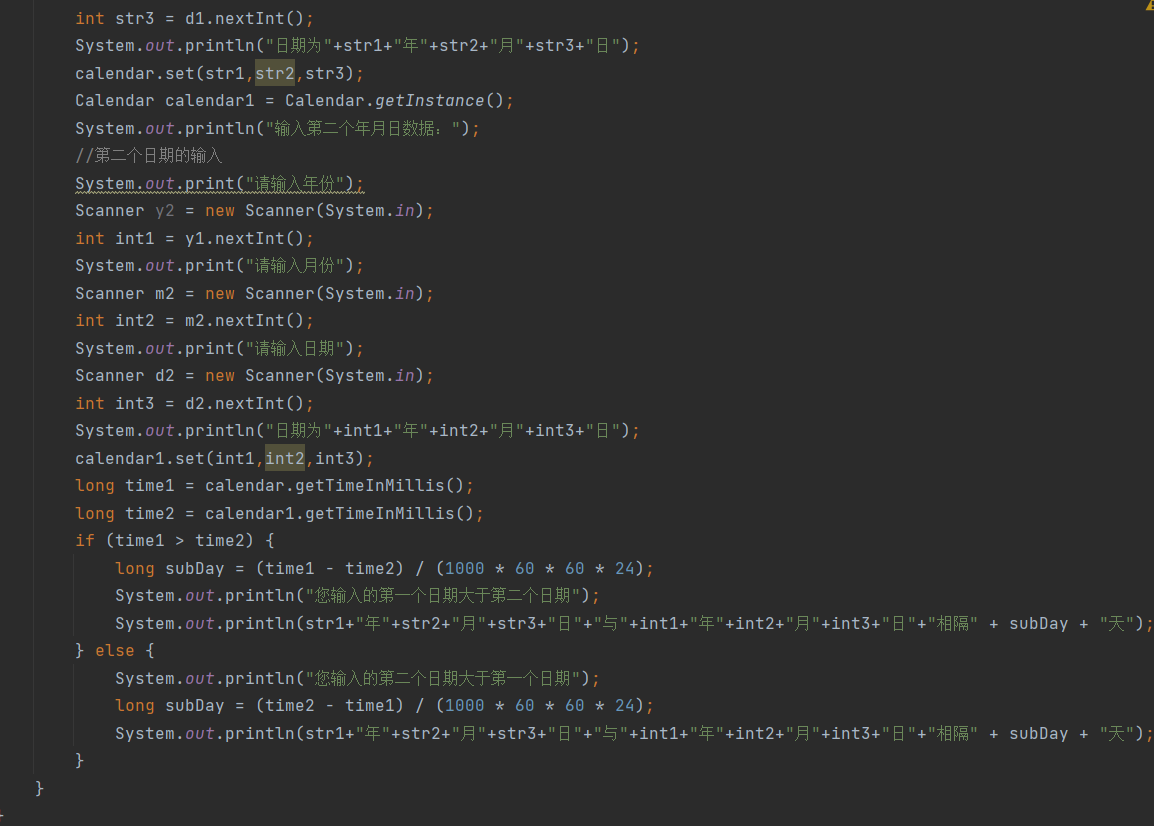
1.主要练习使用Date类和Calendar类。

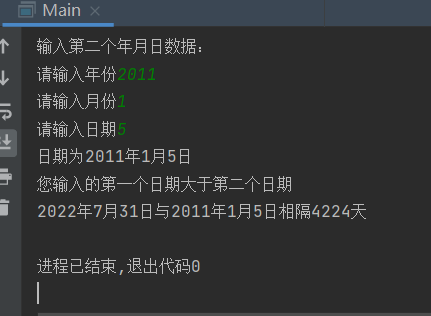
2.问题：编写一个程序，用户输入两个日期，程序将判断两个日期的大小关系，以及两个日期的相隔天数。

3.本次实验主要练习和区分Date类和Calendar类，Date类在java.util包里，使用Date类的无参数构造方法创建的对象可以获取本地当前时间，用Date的构造方法Date（long time）创建的Date对象表示相对于1970年1月1日0点（GMT）的时间。Calendar类在java.util包中。使用Calendar类的static方法getInstance（）可以初始化一个日历对象，然后调用其方法将日历设置到任何一个时间。

### (2)程序的运行和结果

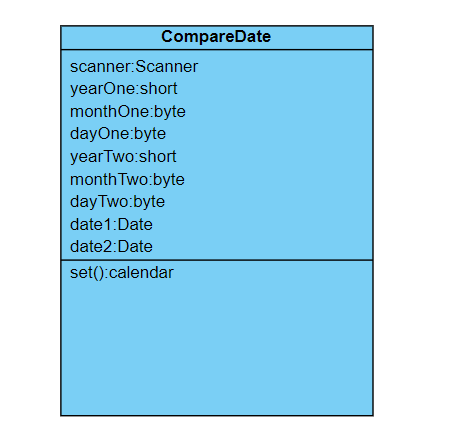




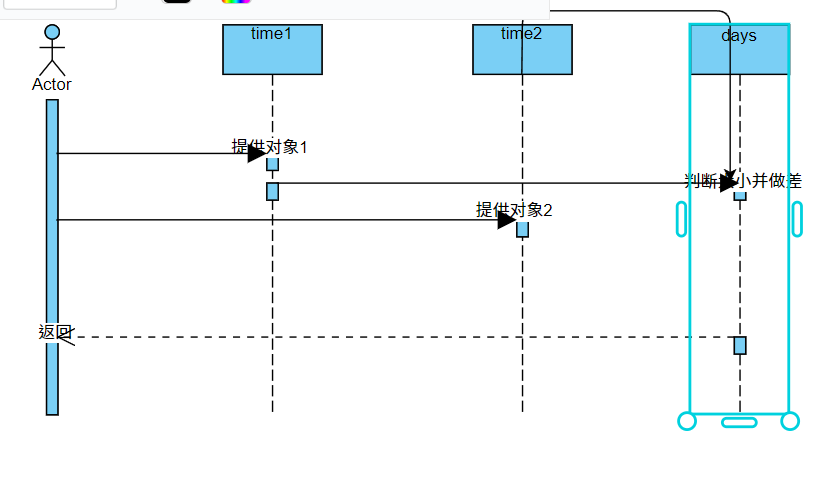


### （3）UML图

类图：



时序图：



### （4）源代码

import java.util.Scanner;  
import java.util.Calendar;  
  
public class Example8\_4 {  
    public static void main(String[] args) {  
        Calendar calendar = Calendar.getInstance();  
        //第一个日期输入  
        System.out.println("输入第一个年月日数据：");  
        System.out.print("请输入年份");  
        Scanner y1 = new Scanner(System.in);  
        int str1 = y1.nextInt();  
        System.out.print("请输入月份");  
        Scanner m1 = new Scanner(System.in);  
        int str2 = m1.nextInt();  
        System.out.print("请输入日期");  
        Scanner d1 = new Scanner(System.in);  
        int str3 = d1.nextInt();  
        System.out.println("日期为"+str1+"年"+str2+"月"+str3+"日");  
        calendar.set(str1,str2,str3);  
        Calendar calendar1 = Calendar.getInstance();  
        System.out.println("输入第二个年月日数据：");  
        //第二个日期的输入  
        System.out.print("请输入年份");  
        Scanner y2 = new Scanner(System.in);  
        int int1 = y1.nextInt();  
        System.out.print("请输入月份");  
        Scanner m2 = new Scanner(System.in);  
        int int2 = m2.nextInt();  
        System.out.print("请输入日期");  
        Scanner d2 = new Scanner(System.in);  
        int int3 = d2.nextInt();  
        System.out.println("日期为"+int1+"年"+int2+"月"+int3+"日");  
        calendar1.set(int1,int2,int3);  
long time1 = calendar.getTimeInMillis();  
        long time2 = calendar1.getTimeInMillis();  
if (time1 > time2) {  
            long subDay = (time1 - time2) / (1000 \* 60 \* 60 \* 24);  
            System.out.println("您输入的第一个日期大于第二个日期");  
            System.out.println(str1+"年"+str2+"月"+str3+"日"+"与"+int1+"年"+int2+"月"+int3+"日"+"相隔" + subDay + "天");  
        } else {  
            System.out.println("您输入的第二个日期大于第一个日期");  
            long subDay = (time2 - time1) / (1000 \* 60 \* 60 \* 24);  
            System.out.println(str1+"年"+str2+"月"+str3+"日"+"与"+int1+"年"+int2+"月"+int3+"日"+"相隔" + subDay + "天");  
        }  
}

4.处理大整数

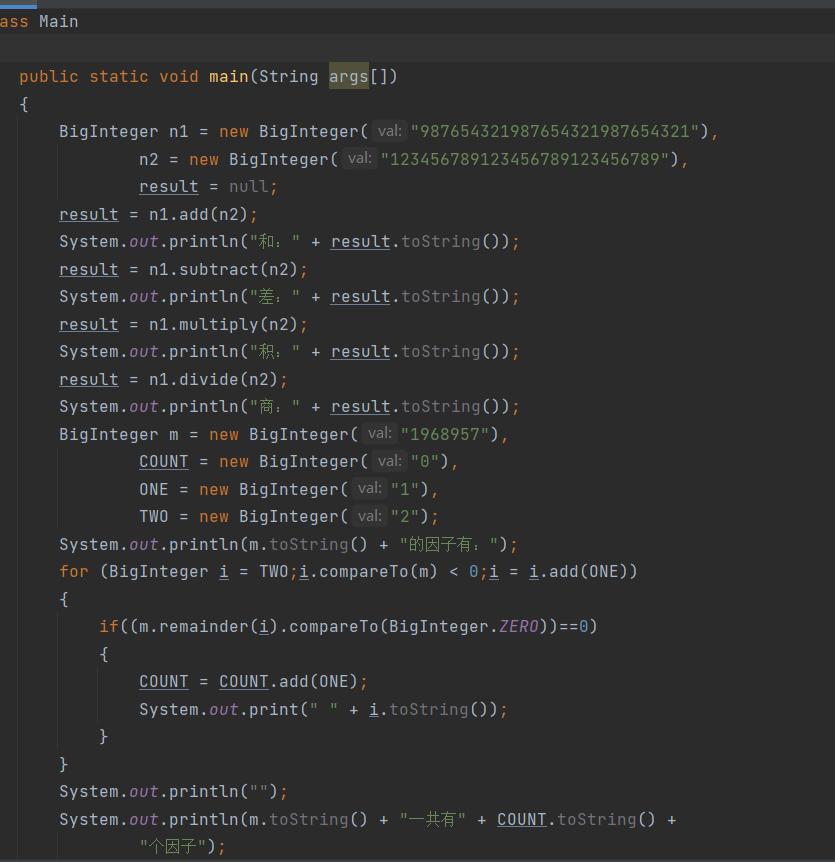
### **（1）问题分析**

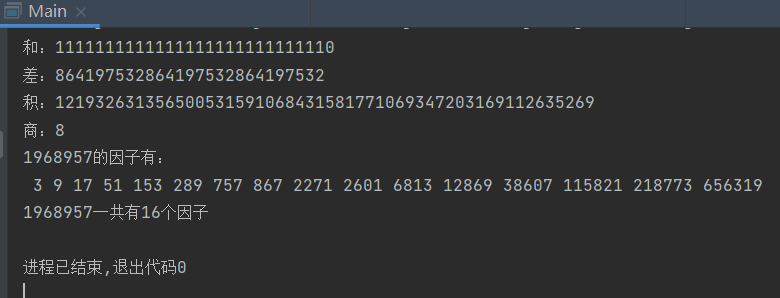
1.实验目的是掌握BigInteger类的常用方法。

2.问题：编写一个程序，计算两个大整数的和、差、积和商，并计算出了一个大整数的因子个数（不包括1和它本身）

3.大整数BigInteger类可以提供任意精度的整数运算，但是其 运算需要借助math包中的计算方法。并且构造方法可以发生NumberFormatException异常。

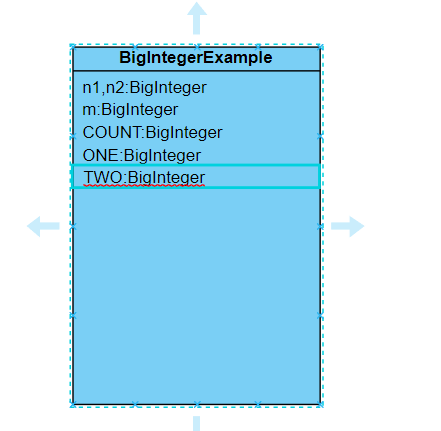
### (2)程序的运行和结果



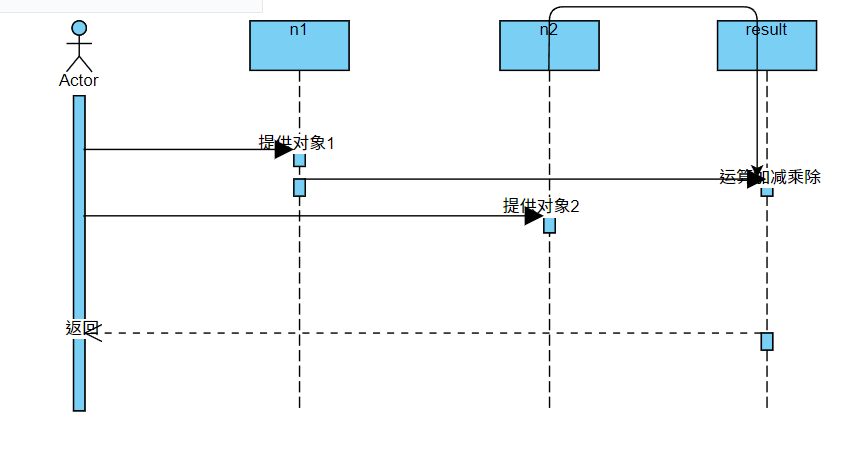


### （3）UML图

类图：



时序图：



### （4）源代码

package com.company;  
import java.math.\*;  
class Main  
{  
    public static void main(String args[])  
    {  
        BigInteger n1 = new BigInteger("987654321987654321987654321"),  
                n2 = new BigInteger("123456789123456789123456789"),  
                result = null;  
        result = n1.add(n2);  
        System.out.println("和：" + result.toString());  
        result = n1.subtract(n2);  
        System.out.println("差：" + result.toString());  
        result = n1.multiply(n2);  
        System.out.println("积：" + result.toString());  
        result = n1.divide(n2);  
        System.out.println("商：" + result.toString());  
        BigInteger m = new BigInteger("1968957"),  
                COUNT = new BigInteger("0"),  
                ONE = new BigInteger("1"),  
                TWO = new BigInteger("2");  
        System.out.println(m.toString() + "的因子有：");  
        for (BigInteger i = TWO;i.compareTo(m) < 0;i = i.add(ONE))  
        {  
            if((m.remainder(i).compareTo(BigInteger.ZERO))==0)  
            {  
                COUNT = COUNT.add(ONE);  
                System.out.print(" " + i.toString());  
            }  
        }  
        System.out.println("");  
        System.out.println(m.toString() + "一共有" + COUNT.toString() +  
                "个因子");  
    }  
}

## 5.替换错别字

### **（1）问题分析**

1.使用Pattern类和Match类检索字符串

2.问题：将错别字替换为正确用字的字符串

3.使用正则表达式regex做参数创建称为模式的Pattern类的实例pattern。然后得到可以检索字符串str的Matcher类的实例matcher，其调用String replaceAll方法返回一个字符串，把str中与模式regex匹配的子字符串全部替换成参数replacement指定的字符串得到的。

### (2)程序的运行和结果

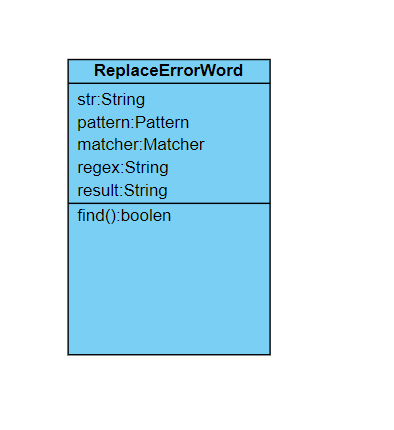




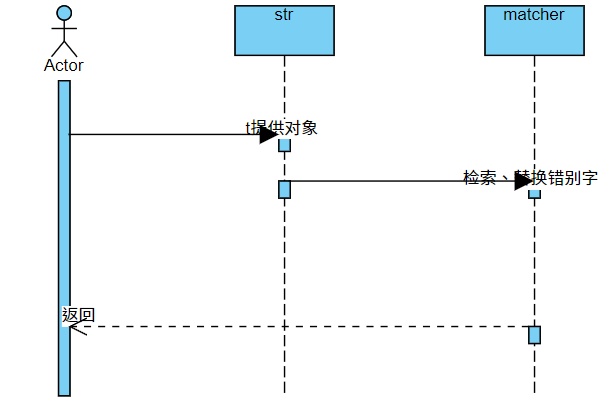


### （3）UML图

类图：



时序图：



### （4）源代码

package com.company;  
  
import java.util.regex.\*;  
  
public class Main {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        String str="忘记密码，不要惊慌失错，请登陆www.yy.cn或登陆www.tt.cc";  
        Pattern pattern1;  
        Pattern pattern2;  
        Matcher matcher1;  
        Matcher matcher2;  
  
        String regex1="登陆";  
        String regex2="惊慌失错";  
  
        pattern1=Pattern.compile(regex1);  
        matcher1=pattern1.matcher(str);  
  
        while(matcher1.find()){  
            String s=matcher1.group(); //返回find（）找到的子串  
            System.out.print(s+"\n");  
        }  
  
        System.out.print("将\"登陆\"替换为\"登录\"的字符串："+"\n");  
        String result=matcher1.replaceAll("登录");  
        System.out.print(result+"\n");  
  
  
        pattern2=Pattern.compile(regex2);  
        matcher2=pattern2.matcher(result);  
  
        while(matcher2.find()){  
            String s=matcher2.group(); //返回find（）找到的子串  
           // System.out.print(matcher2.start()+"位置出现：");  
            System.out.print(s+"\n");  
        }  
        System.out.print("将\"惊慌失错\"替换为\"惊慌失措\"的字符串："+"\n");  
        result=matcher2.replaceAll("惊慌失措");  
        System.out.println(result);  
        System.out.println("20211120138薛凡豪");  
  
    }  
  
}

二、总结

1. 本次实验运用了常用实用类来解决问题。

2.首先使用了String类，String类是一个final类，要注意，其不可以有子类。

3. 第二个是Scanner类，学习使用正则表达式用法来解析字符串得到数据。

4.第三个是日期类（Date类和Calendar类）的使用，注意Date类有无参数的区别。。

5.第四个是大整数Integer类，要注意的是，大整数不可以直接进行四则运算，必须使用其特定的算法（方法）。

6.第五个是Pattern类和Match类，其可以检索字符串，并且做替换等操作。