《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》实验十

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.分析成绩单

## （1）问题分析：

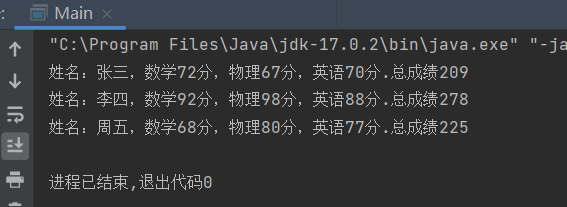
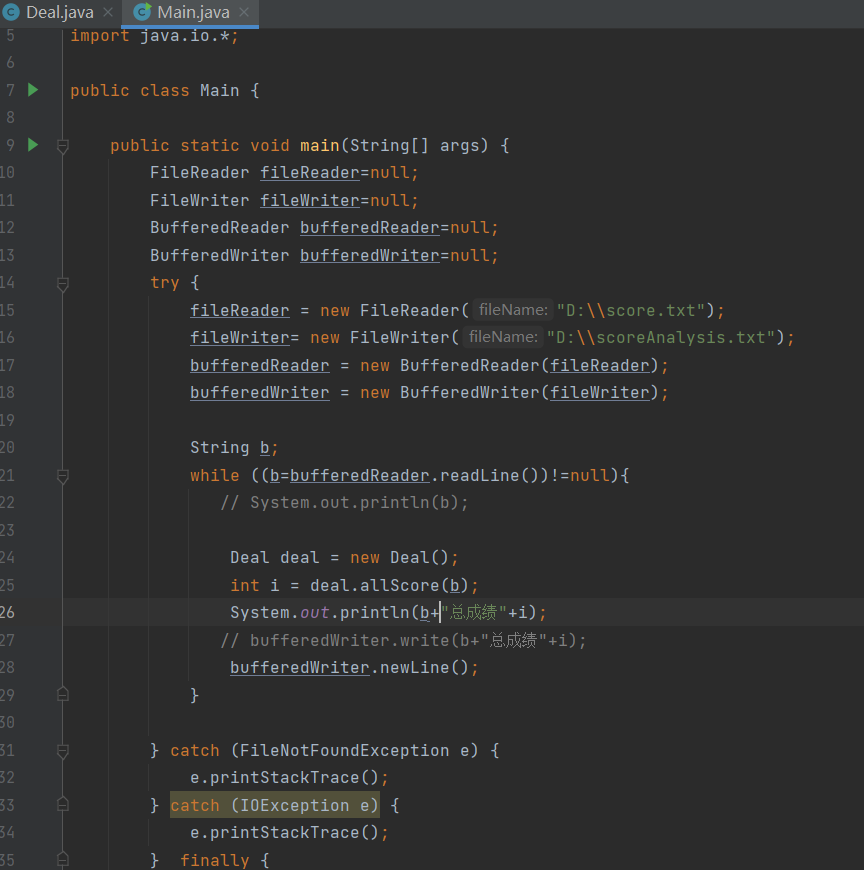
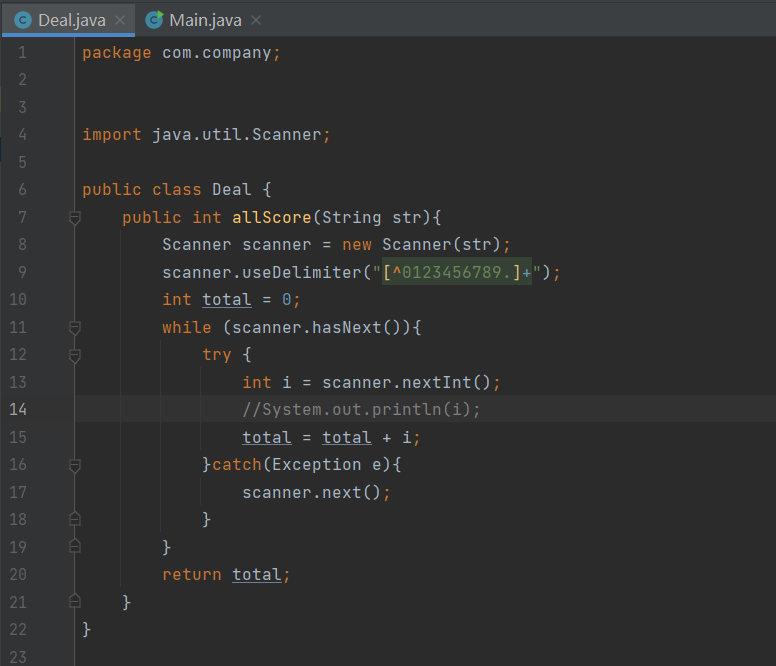
1.本实验主要需要运用的是输入和输出流（IO）

2.重点是掌握文件的读取和输出和字符串的分割以及读取想要的数据。

3.编写程序，两个类组成，一个类给出输入和输出流，负责数据的处理，另一个类为运行类并进行错误抛出，提供总成绩。

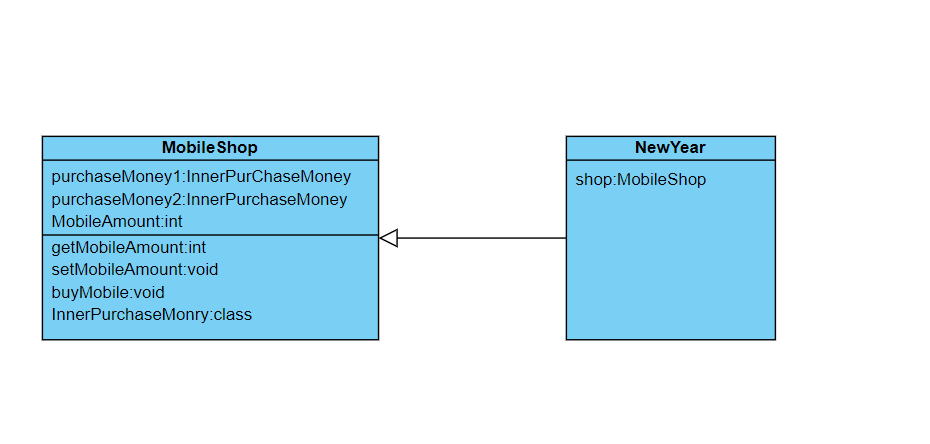
4.本实验用到了字符串的分割以及输入输出流等方法。

### （2）程序的运行和结果

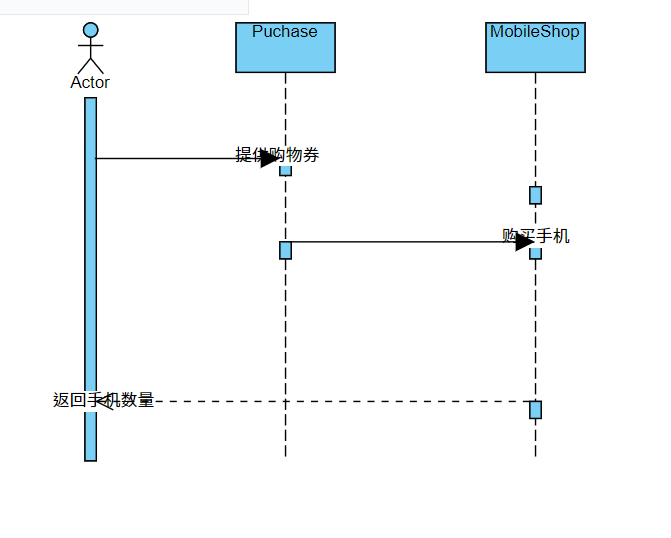


### （3）UML图

类图：



时序图：



### （4）源代码：

package com.company;  
  
  
import java.util.Scanner;  
  
public class Deal {  
    public int allScore(String str){  
        Scanner scanner = new Scanner(str);  
        scanner.useDelimiter("[^0123456789.]+");  
        int total = 0;  
        while (scanner.hasNext()){  
            try {  
                int i = scanner.nextInt();  
                  
                total = total + i;  
            }catch(Exception e){  
                scanner.next();  
            }  
        }  
        return total;  
    }  
}  
  
  
  
package com.company;  
  
  
  
import java.io.\*;  
  
public class Main {  
  
    public static void main(String[] args) {  
        FileReader fileReader=null;  
        FileWriter fileWriter=null;  
        BufferedReader bufferedReader=null;  
        BufferedWriter bufferedWriter=null;  
        try {  
            fileReader = new FileReader("D:\\score.txt");  
            fileWriter= new FileWriter("D:\\scoreAnalysis.txt");  
            bufferedReader = new BufferedReader(fileReader);  
            bufferedWriter = new BufferedWriter(fileWriter);  
  
            String b;  
            while ((b=bufferedReader.readLine())!=null){  
  
  
                Deal deal = new Deal();  
                int i = deal.allScore(b);  
                System.out.println(b+"总成绩"+i);  
                bufferedWriter.write(b+"总成绩"+i);  
                bufferedWriter.newLine();  
            }  
  
        } catch (FileNotFoundException e) {  
            e.printStackTrace();  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  finally {  
            try {  
                fileReader.close();  
                bufferedWriter.flush();  
                fileWriter.close();  
                bufferedReader.close();  
                bufferedWriter.close();  
            } catch (IOException e) {  
                e.printStackTrace();  
            }  
  
        }  
  
  
    }

## 2.统计英文单词

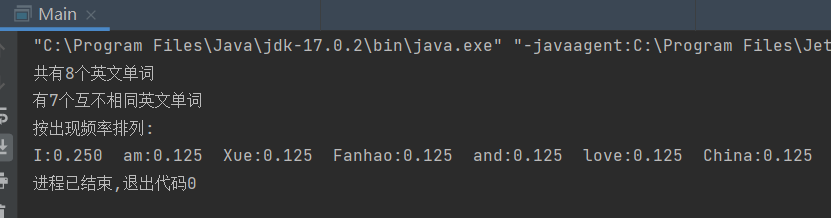
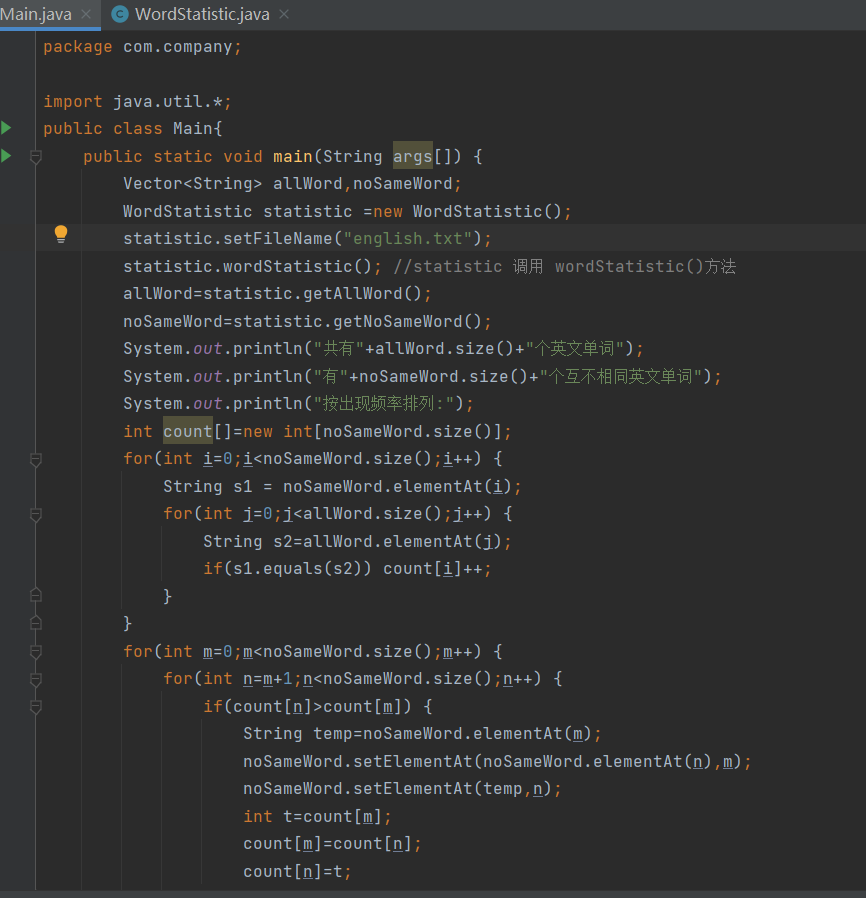
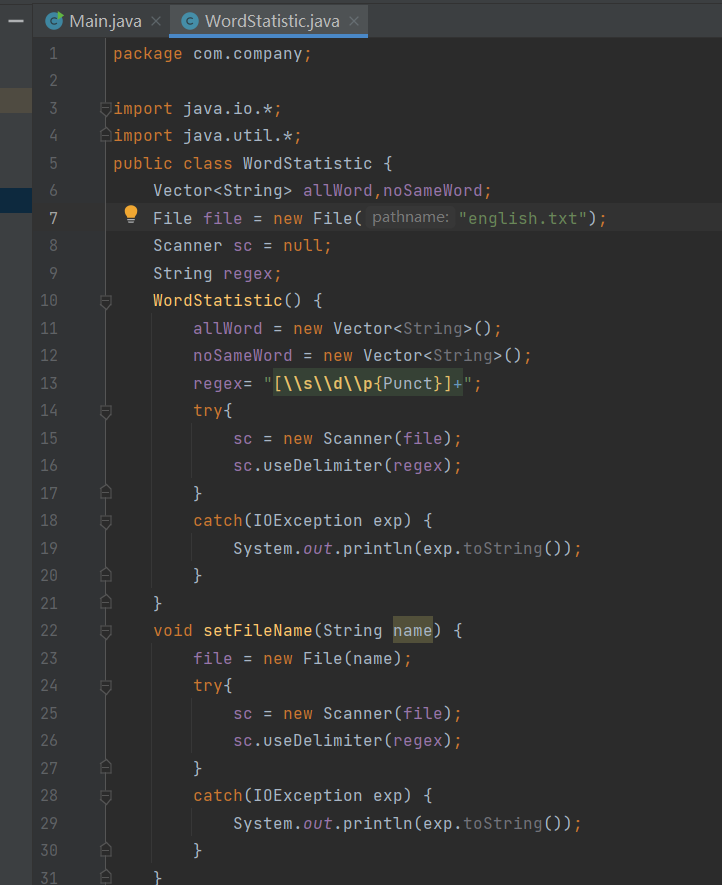
### （1）问题分析：

1.本次实验主要练习输入输出流以及正则表达式的使用。

2.问题：编写一个程序，读取指定文件，并输出英文单词的总数和多少互不相同的单词，并统计不同单词的出现频率。

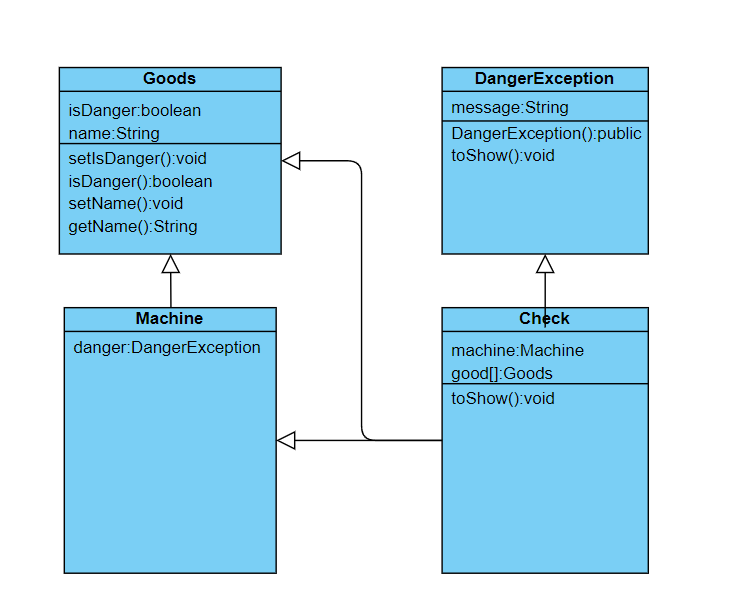
3.实验主要运用了输入输出流正则表达式和文件的读取。

### （2）程序的运行和结果

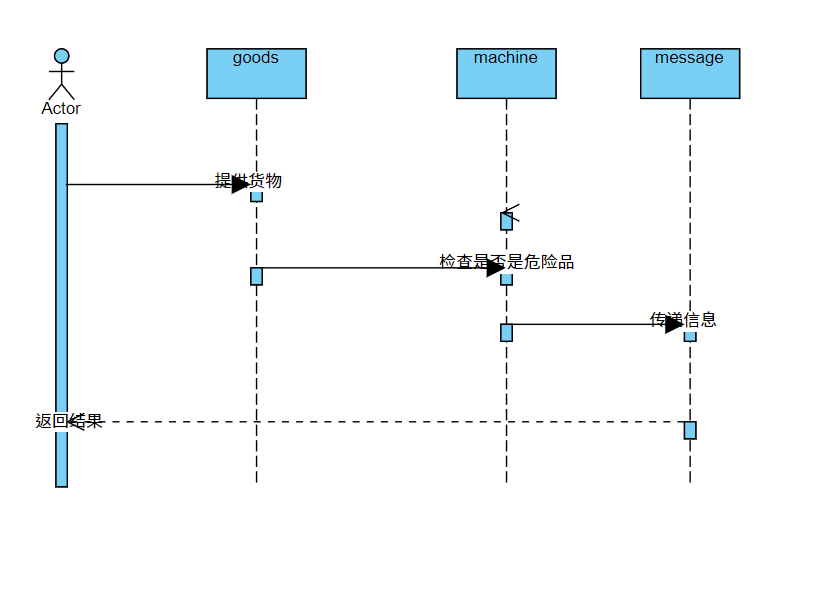


### （3）UML图

类图：



时序图：



### （4）源代码

package com.company;  
  
import java.io.\*;  
import java.util.\*;  
public class WordStatistic {  
    Vector<String> allWord,noSameWord;  
    File file = new File("english.txt");  
    Scanner sc = null;  
    String regex;  
    WordStatistic() {  
        allWord = new Vector<String>();  
        noSameWord = new Vector<String>();  
        regex= "[\\s\\d\\p{Punct}]+";  
        try{  
            sc = new Scanner(file);  
            sc.useDelimiter(regex);  
        }  
        catch(IOException exp) {  
            System.out.println(exp.toString());  
        }  
    }  
    void setFileName(String name) {  
        file = new File(name);  
        try{  
            sc = new Scanner(file);  
            sc.useDelimiter(regex);  
        }  
        catch(IOException exp) {  
            System.out.println(exp.toString());  
        }  
    }  
    public void wordStatistic() {  
        try{  
            while(sc.hasNext()){  
                String word = sc.next();  
                allWord.add(word);  
                if(!noSameWord.contains(word))  
                    noSameWord.add(word);  
            }  
        }  
        catch(Exception e){}  
    }  
    public Vector<String> getAllWord() {  
        return allWord;  
    }  
    public Vector<String> getNoSameWord() {  
        return noSameWord;  
    }  
}  
package com.company;  
  
import java.util.\*;  
public class Main{  
    public static void main(String args[]) {  
        Vector<String> allWord,noSameWord;  
        WordStatistic statistic =new WordStatistic();  
        statistic.setFileName("english.txt");  
        statistic.wordStatistic(); /atistic 调用 wordStatistic()方法  
        allWord=statistic.getAllWord();  
        noSameWord=statistic.getNoSameWord();  
        System.out.println("共有"+allWord.size()+"个英文单词");  
        System.out.println("有"+noSameWord.size()+"个互不相同英文单词");  
        System.out.println("按出现频率排列:");  
        int count[]=new int[noSameWord.size()];  
        for(int i=0;i<noSameWord.size();i++) {  
            String s1 = noSameWord.elementAt(i);  
            for(int j=0;j<allWord.size();j++) {  
                String s2=allWord.elementAt(j);  
                if(s1.equals(s2)) count[i]++;  
            }  
        }  
        for(int m=0;m<noSameWord.size();m++) {  
            for(int n=m+1;n<noSameWord.size();n++) {  
                if(count[n]>count[m]) {  
                    String temp=noSameWord.elementAt(m);  
                    noSameWord.setElementAt(noSameWord.elementAt(n),m);  
                    noSameWord.setElementAt(temp,n);  
                    int t=count[m];  
                    count[m]=count[n];  
                    count[n]=t;  
                }  
            }  
        }  
        for(int m=0;m<noSameWord.size();m++) {  
            double frequency=(1.0\*count[m])/allWord.size();  
            System.out.printf("%s:%-7.3f",noSameWord.elementAt(m),frequency);  
        }  
    }  
}  
}3.读取压缩文件

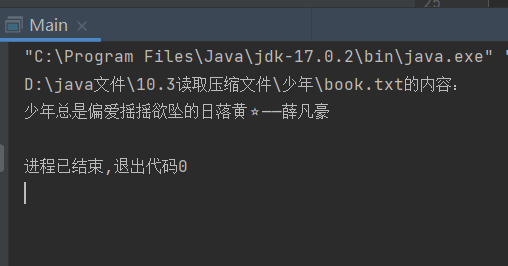
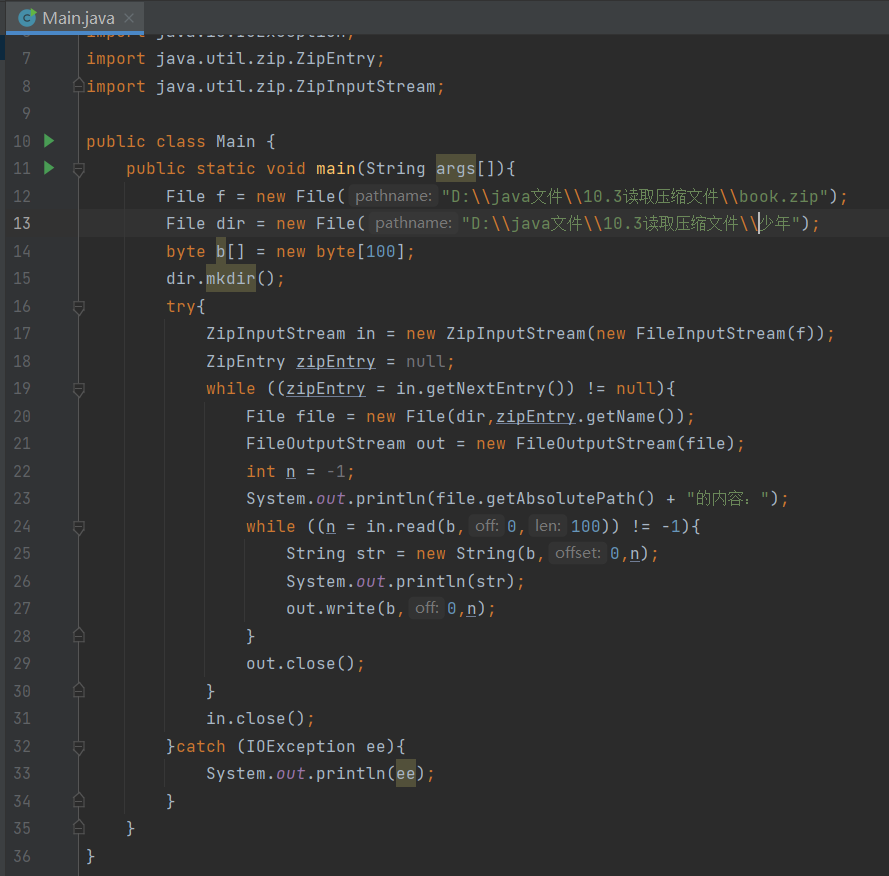
### (1)问题分析：

1.主要练习使用ZipInputStream流进行解压缩和文件读取。

2.问题：编写一个程序，解压缩指定文件，并将其中含有的文件重新存放到当前目录内。即解压缩至另一个文件夹。并读取内容。

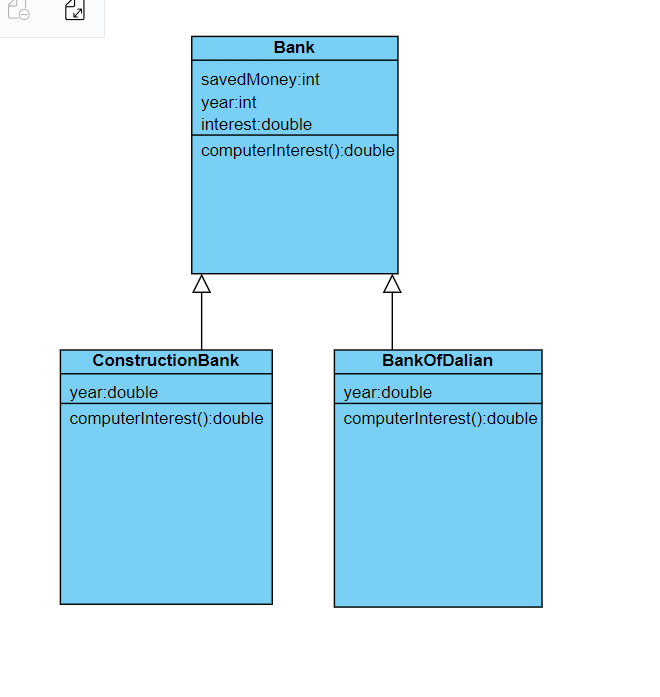
3.本次实验主要练习输入输出流和ZipInputStream流。

### (2)程序的运行和结果

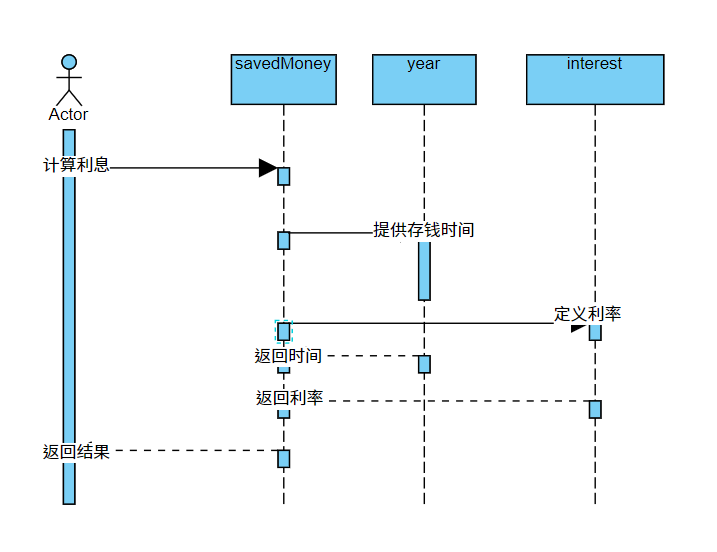


### （3）UML图

类图：



时序图：



### （4）源代码

package com.company;  
  
import java.io.File;  
import java.io.FileInputStream;  
import java.io.FileOutputStream;  
import java.io.IOException;  
import java.util.zip.ZipEntry;  
import java.util.zip.ZipInputStream;  
  
public class Main {  
    public static void main(String args[]){  
        File f = new File("D:\\java文件\\10.3读取压缩文件\\book.zip");  
        File dir = new File("D:\\java文件\\10.3读取压缩文件\\少年");  
        byte b[] = new byte[100];  
        dir.mkdir();  
        try{  
            ZipInputStream in = new ZipInputStream(new FileInputStream(f));  
            ZipEntry zipEntry = null;  
            while ((zipEntry = in.getNextEntry()) != null){  
                File file = new File(dir,zipEntry.getName());  
                FileOutputStream out = new FileOutputStream(file);  
                int n = -1;  
                System.out.println(file.getAbsolutePath() + "的内容：");  
                while ((n = in.read(b,0,100)) != -1){  
                    String str = new String(b,0,n);  
                    System.out.println(str);  
                    out.write(b,0,n);  
                }  
                out.close();  
            }  
            in.close();  
        }catch (IOException ee){  
            System.out.println(ee);  
        }  
    }  
}

二、总结

1. 本次实验主要运用了输入输出流来解决问题。

2.首先使用了读取指定文件来实现具体的问题。

3. 第二个使用字符的输入输出流，学习使用字符串分割用法来进行数据提取分析。

4.第三个是Scanner来解析文件，进行数据分析。。

5.第四个是ZipInputStream流的使用，对压缩文件进行解压缩和读取。

6.本次实验主要运用java的输入输出流用法，包括文件字符的解析还有Scanner解析文件，还包括了压缩文件的解压缩和读取问题。