电子科技大学信息与软件工程学院

实 验 报 告

（四）

学　　号 2018091618008

姓　　名 袁昊男

（实验）课程名称 面向对象程序设计 Java

理论教师 周帆

实验教师 何中海

电 子 科 技 大 学

实　验　报　告

学生姓名：袁昊男　　学号：2018091618008　　指导教师：周帆

实验地点：信软楼 304　　实验时间：2019.11.24、12.01

一、 实验名称：Java 高级编程技术

二、 实验学时：2 学时

三、 实验目的：

理解 Java 异常处理、多线程、I/O、网络以及 GUI 等高级编程技术。

四、 实验原理：

1、 面向对象

Java 语言具有真正的面向对象语言的特点，除了数值、布尔和字符三种基本的数据类型外，其它类型都是对象。它支持封装、多态性和继承。封装就是将对象内的数据和代码联编起来，形成一个对象；多态性是指一个接口，有多个内在实现形式表示；继承是指某一对象直接使用另一对象的所有属性和方法的过程。

对程序员来说，这意味着要注意应中的数据和操纵数据的方法（method），而不是严格地用过程来思考。在一个面向对象的系统中，类（class）是数据和操作数据的方法的集合。数据和方法一起描述对象（object）的状态和行为。每一对象是其状态和行为的封装。类是按一定体系和层次安排的，使得子类可以从超类继承行为。在这个类层次体系中有一个根类，它是具有一般行为的类。Java 程序是用类来组织的。

2、 编译和解释性

Java 编译程序生成字节码（byte-code），而不是通常的机器码。Java 字节码提供对体系结构中性的目标文件格式，代码设计成可有效地传送程序到多个平台。Java 程序可以在任何实现了 Java 解释程序和运行系统（run-time system）的系统上运行。

在一个解释性的环境中，程序开发的标准“链接”阶段大大消失了。如果说 Java 还有一个链接阶段，它只是把新类装进环境的过程，它是增量式的、轻量级的过程。因此，Java 支持快速原型和容易试验，

它将导致快速程序开发。这是一个与传统的、耗时的“编译、链接和测试”形成鲜明对比的精巧的开发过程。

五、 实验内容：

1、 完成第七章习题 6 编程；
2、 完成第八章习题 6 编程；
3、 完成第九章习题 10 编程；
4、 完成第十一章习题 7 编程；
5、 完成第十一章习题 10 编程。

六、 实验器材（设备、元器件）：

配置了 JDK 环境、安装有 Eclipse 软件的个人电脑一台。

七、 实验步骤：

1、 按照题目要求分析需求、功能；
2、 设计算法，编写程序并进行测试；
3、 结果分析，撰写实验报告。

八、 实验结果与分析（含重要数据结果分析或核心代码流程分析）

1、 第七章习题 6：

编写一个含有 ArithmeticException、IndexOutBoundsException 和 NullPointerException 异常处理程序。

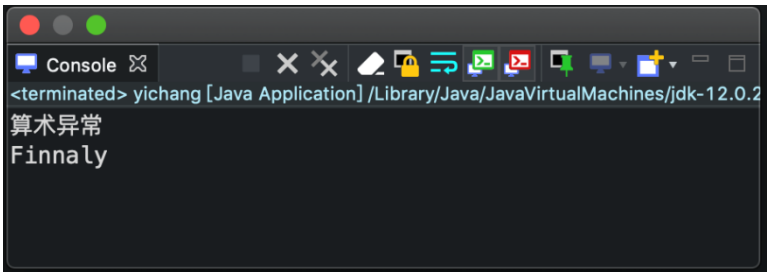
(1) 代码

```
1. package pkg;
2.
3. public class yichang {
4.     public static void main(String[] args) {
5.         try {
6. //           System.out.println(1 / 0);
7. //           System.out.println(new int[] {}[0]);
8. //           String str = null;
9. //           System.out.println(str.toString());
10.        }
11.        catch (ArithmeticException e) {
12.            System.out.println("算术异常");
13.        }
14.        catch (ArrayIndexOutOfBoundsException e) {
15.            System.out.println("数组下标越界异常");
16.        }
17.        catch (NullPointerException e) {
18.            System.out.println("空指针异常");
19.        }
20.        finally {
21.            System.out.println("Finnaly");
22.        }
```

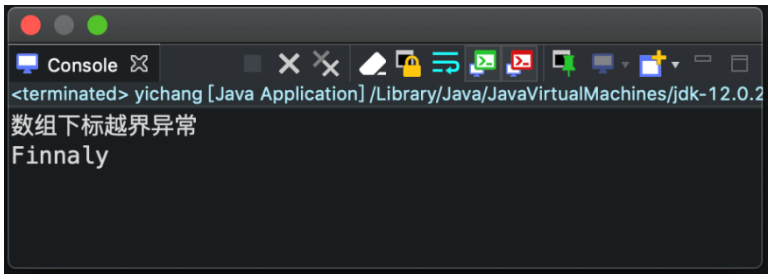
```
    }
}
```

(2) 运行截图

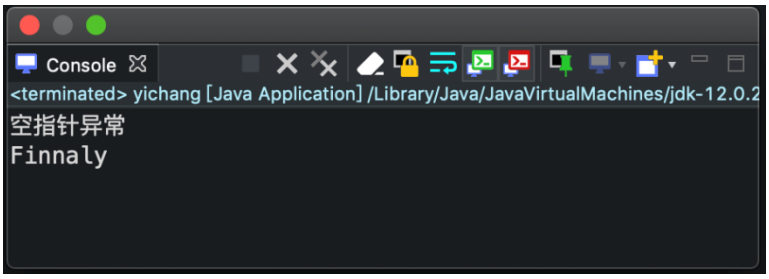
a) ArithmeticException 算术异常



b) IndexOutBoundsException 数组下标越界异常



c) NullPointerException 空指针异常



2、 第八章习题 6：

利用多线程编写一个可以同时显示 3 个不同时区的时钟程序。

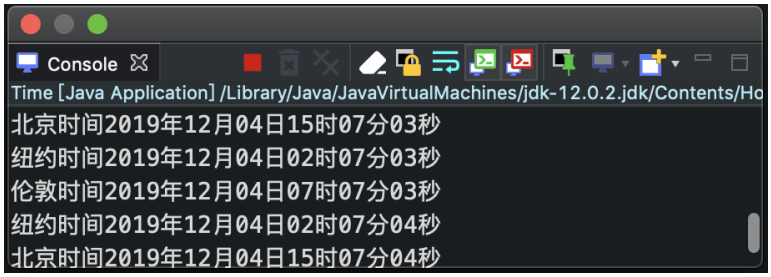
(1) 代码

```
package pkg2;

import java.util.Date;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Locale;

public class Time {
    public Time() {
        FirstThread first = new FirstThread();
        SecondThread second = new SecondThread();
        ThirdThread third = new ThirdThread();
        Thread thread1 = new Thread(first);
        Thread thread2 = new Thread(second);
        Thread thread3 = new Thread(third);
```

```
16.         thread1.setPriority(Thread.MIN_PRIORITY);
17.         thread2.setPriority(Thread.MAX_PRIORITY);
18.         thread3.setPriority(Thread.MIN_PRIORITY);
19.         thread1.start();
20.         thread2.start();
21.         thread3.start();
22.
23.     }
24.
25.     public static void main(String[] args) {
26.         new Time();
27.     }
28. }
29.
30. class FirstThread implements Runnable {
31.     public void run() {
32.         try {
33.             for (int i = 0; i < 100; i++) {
34.                 SimpleDateFormat sdf = new Sim-
    pleDateFormat("北京时间 yyyy 年 MM 月 dd 日 HH 时 mm 分 ss 秒
    ");// 格式化输出日期
35.                 Date now = new Date();
36.                 System.out.println(sdf.for-
    mat(now));
37.                 Thread.sleep(1000);
38.                 Thread.yield();
39.             }
40.         } catch (InterruptedException e) {
41.         }
42.     }
43. }
44.
45. class SecondThread implements Runnable {
46.     public void run() {
47.         try {
48.             for (int i = 0; i < 100; i++) {
49.                 SimpleDateFormat sdf = new Sim-
    pleDateFormat("伦敦时间 yyyy 年 MM 月 dd 日 HH 时 mm 分 ss 秒
    ");// 格式化输出日期
50.                 Date now = new Date();
51.                 long time = 60 * 1000;// 60 秒
52.                 Date beforeDate = new Date(now.get-
    Time() - time * 480);// 60 秒前的时间
53.                 System.out.println(sdf.format(be-
    foreDate));
54.                 Thread.sleep(1000);
55.                 Thread.yield();
56.             }
57.         } catch (InterruptedException e) {
58.         }
59.     }
60. }
61.
62. class ThirdThread implements Runnable {
63.     public void run() {
64.         try {
65.             for (int i = 0; i < 100; i++) {
```

```
66.                 SimpleDateFormat sdf = new Sim-
   pleDateFormat("纽约时间 yyyy 年 MM 月 dd 日 HH 时 mm 分 ss 秒
   ");// 格式化输出日期
67.                 Date now = new Date();
68.                 long time = 60 * 1000;// 60 秒
69.                 Date beforeDate = new Date(now.get-
   Time() - time * 780);// 60 秒前的时间
70.                 System.out.println(sdf.format(be-
   foreDate));
71.                 Thread.sleep(1000);
72.                 Thread.yield();
73.             }
74.         } catch (InterruptedException e) {
75.         }
76.     }
77. }
```

(2) 运行截图



3、 第九章习题 10：

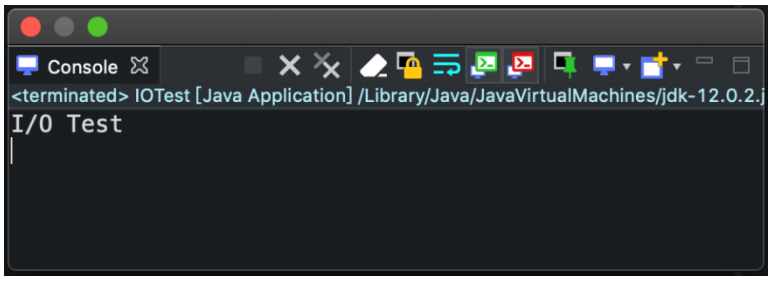
编程实现从键盘读入数据，保存到指定的文件中。

(1) 代码

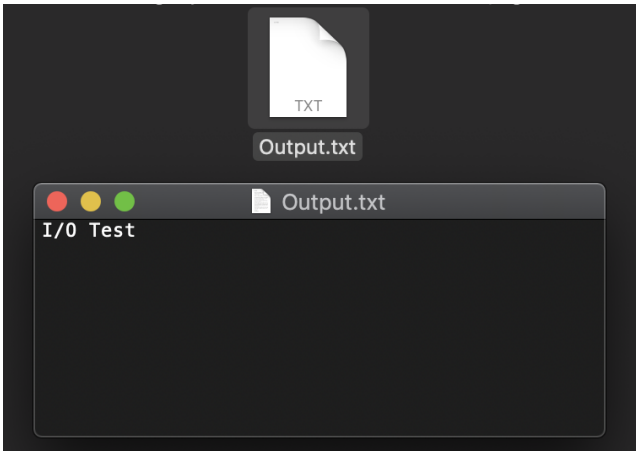
```
1.  package pkg2;
2.
3.  import java.io.BufferedWriter;
4.  import java.io.File;
5.  import java.io.FileWriter;
6.  import java.io.IOException;
7.  import java.util.Scanner;
8.  public class IOTest {
9.      pub-
   lic static void main(String[] args) throws IOExcep-
   tion {
10.         File ofile=new File("src/Output.txt");
11.         if (!ofile.isFile()) {
12.             ofile.createNewFile();
13.         }
14.         BufferedWriter writer = new Buff-
   eredWriter(new FileWriter("src/Output.txt"));
15.         Scanner scanner=new Scanner(System.in);
16.         String input=scanner.nextLine();
17.         writer.write(input);
18.         writer.close();
19.         scanner.close();
20.     }
21. }
```

(2) 运行截图

a) 从键盘读入数据



b) 保存到指定文件



4、 第十一章习题 7：

请编写一个程序在面板上显示三个按钮，按钮上的标签分别是：set red、set green、set blue。

✧ 当按下 set red 按钮，则三个按钮的标签都变成红色；

✧ 按下 set green 按钮则变成绿色；

✧ 按下 set blue 按钮则变成蓝色。

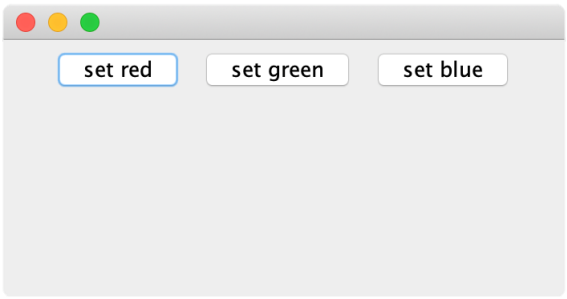
(1) 代码

```
1.  package pkg2;
2.
3.  import java.awt.Color;
4.  import java.awt.event.ActionEvent;
5.  import java.awt.event.ActionListener;
6.  import javax.swing.*;
7.  public class App extends JFrame implements Action-
    Listener {
8.      // 实现的点击按钮
9.      JButton jb1, jb2, jb3;
10.     public App() {
11.         JFrame frm = new JFrame();
12.         jb1 = new JButton("set red");
13.         jb2 = new JButton("set green");
14.         jb3 = new JButton("set blue");
15.         JPanel jp = new JPanel();
16.         jp.add(jb1);
17.         jp.add(jb2);
```

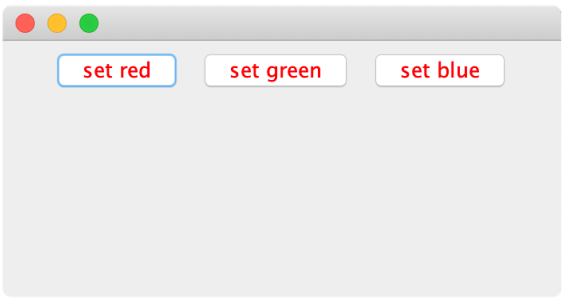
```
18.         jp.add(jb3);
19.         this.add(jp);
20.         this.setVisible(true);
21.         this.setDefaultCloseOperation(3);
22.         this.setSize(500, 500);
23.         // 需要对按钮就行监听
24.         jb1.addActionListener(this);// 这里需要指的是
    当前类
25.         jb2.addActionListener(this);
26.         jb3.addActionListener(this);
27.     }
28.     public void actionPerformed(ActionEvent e) {
29.         if (e.getSource() == jb1) {
30.             jb1.setForeground(Color.RED);
31.             jb2.setForeground(Color.RED);
32.             jb3.setForeground(Color.RED);
33.         } else if (e.getSource() == jb2) {
34.             jb1.setForeground(Color.GREEN);
35.             jb2.setForeground(Color.GREEN);
36.             jb3.setForeground(Color.GREEN);
37.         } else if (e.getSource() == jb3) {
38.             jb1.setForeground(Color.BLUE);
39.             jb2.setForeground(Color.BLUE);
40.             jb3.setForeground(Color.BLUE);
41.         }
42.     }
43.     public static void main(String[] args) {
44.         new App();
45.     }
46. }
```

(2) 运行截图

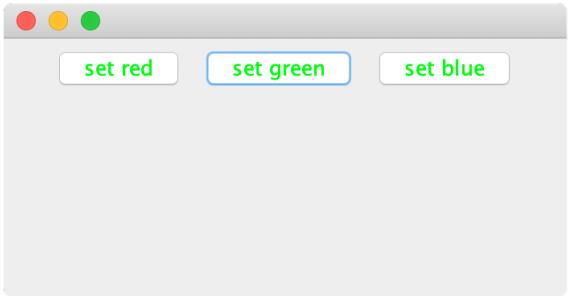
a) 初始化



b) 按下 set red



c) 按下 set green



d) 按下 set blue

5、 第十一章习题 10：

编写一个类似 Windows“记事本”的界面程序。

(1) 代码

```
1. package pkg2;
2.
3. import java.awt.Color;
4. import java.awt.event.ActionEvent;
5. import java.awt.event.ActionListener;
6. import java.io.BufferedReader;
7. import java.io.BufferedWriter;
8. import java.io.FileReader;
9. import java.io.FileWriter;
10.
11.import javax.swing.JFileChooser;
12.import javax.swing.JFrame;
13.import javax.swing.JMenu;
14.import javax.swing.JMenuBar;
15.import javax.swing.JMenuItem;
16.import javax.swing.JTextArea;
17.//让其继承窗口类
18.public class NotePal extends JFrame implements Ac-
   tionListener {
19.
20.    private static final long serialVer-
   sionUID = 1L;    //定义一个文本框
21.    JTextArea jTextArea = null;     //定义一个菜单栏
22.    JMenuBar jMenuBar = null;   //定义一个菜单
23.    JMenu jMenu1 = null;    //定义一个“打开”子选项
24.    JMenuItem jMenuItem1 = null;    //定义一个“保存”子
   选项
```

```
25.     JMenuItem jMenuItem2 = null;    //定义一个文件选
    择
26.     JFileChooser jFileChooser = null;   //定义一个
    FileReader 文件输入流
27.     FileReader fileReader = null;   //定义一个 File-
    Writer 输出流
28.     FileWriter fileWriter = null;   //定义一个缓冲字符
    输入流
29.     BufferedReader bufferedReader = null;   //定义一
    个缓冲字符输出流
30.     BufferedWriter bufferedWriter = null;
31. //  @SuppressWarnings("unused")
32.     public static void main(String[] args) {
33.         NotePal notePal = new NotePal();
34.     }
35.
36.     public NotePal()
37.     {
38.         jTextArea = new JTextArea();
39.         jMenuBar = new JMenuBar();
40.         jMenu1 = new JMenu("文件");
41.         jMenuItem1 = new JMenuItem("打开");
42.         jMenuItem1.addActionListener(this);
43.         jMenuItem1.setActionCommand("打开");
44.         jMenuItem2 = new JMenuItem("保存");
45.         jMenuItem2.addActionListener(this);
46.         jMenuItem2.setActionCommand("保存");
47.         jTextArea.setBackground(Color.LIGHT_GRAY);
48.
49.         //将组件添加上各自的位置
50.         //将 jMenuBar 添加到 JFrame 中
51.         this.setJMenuBar(jMenuBar);
52.         //将 jMenu 添加到 jMenuBar 中
53.         jMenuBar.add(jMenu1);
54.         //将 jMenuTItem1 添加到 Jmenu1 中
55.         jMenu1.add(jMenuItem1);
56.         //将 jMenuTItem2 添加到 Jmenu1 中
57.         jMenu1.add(jMenuItem2);
58.         //将 jTextArea 添加到 JFrame 中
59.         this.add(jTextArea);
60.
61.         //设置窗体的标题
62.         this.setTitle("记事本");
63.         //设置窗体的大小
64.         this.setSize(600,400);
65.         //当关闭窗口的时候，关闭进程
66.         this.setDefaultCloseOpera-
    tion(JFrame.EXIT_ON_CLOSE);
67.         //显示窗口
68.         this.setVisible(true);
69.     }
70.     @Override
71.     public void actionPerformed(ActionEvent e) {
72.         if(e.getActionCommand().equals("打开"))
73.         {
74.             //实例化一个 JFileChoose
75.             jFileChooser = new JFileChooser();
76.             //设置文件选择窗口的名字
```

```
77.             jFileChooser.setDialogTitle("选择文
    件...");
78.             //设置文件窗口的默认路径
79.             jFileChooser.showOpenDialog(null);
80.             //显示文件窗口
81.             jFileChooser.setVisible(true);
82.             //用 address 保存用户编辑文件的绝对路径
83.                 String address = jFileChooser.get-
    SelectedFile().getAbsolutePath();
84.             try {
85.                 //为 fileReader 分配空间
86.                 fileReader = new FileReader(ad-
    dress);
87.                 //为 bufferedReader 分配空间
88.                 bufferedReader = new Buff-
    eredReader(fileReader);
89.                 //定义一个 str 判断输入的字符是否已经为
    空
90.                 String str = "";
91.                 //定义一个 strAll 接收文件的全部信息
92.                 String strAll = "";
93.                 //读取文件信息，并将其保存到 strAll 中
94.                 while((str = bufferedReader.read-
    Line()) != null)
95.                 {
96.                     strAll += str + "\r\n";
97.                 }
98.                 //将 strAll 中的全部信息显示到 JTextArea
    上
99.                 jTextArea.setText(strAll);
100.            } catch (Exception e2) {
101.                e2.printStackTrace();
102.            }finally{
103.                try {
104.                    //关闭文件
105.                    bufferedReader.close();
106.                    fileReader.close();
107.                } catch (Exception e3) {
108.                    e3.printStackTrace();
109.                }
110.            }
111.        }
112.        //如果用户点的是保存按钮
113.        if(e.getActionCommand().equals("保存"))
114.        {
115.            //创建一个保存窗口
116.            JFileChooser jFileChooser1 = new JFile-
    Chooser();
117.            //设置窗口名字
118.            jFileChooser1.setDialogTitle("另存
    为...");
119.            //设置默认设置
120.            jFileChooser1.showSaveDialog(null);
121.            //显示窗口
122.            jFileChooser1.setVisible(true);
123.            try {
124.                //为 fileWrite 分配空间
```

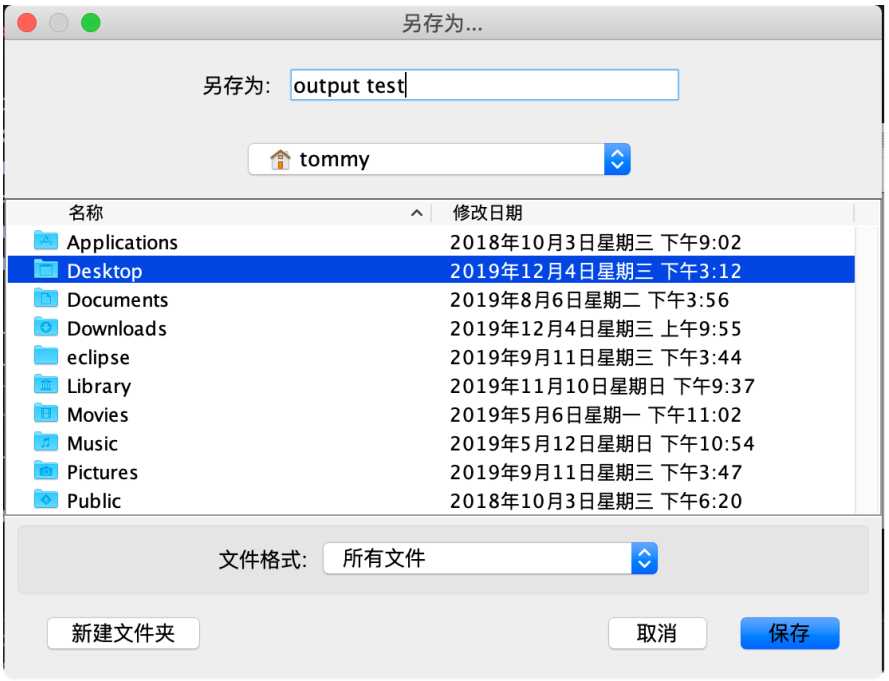
```
125.                 fileWriter = new FileWriter(jFile-
    Chooser1.getSelectedFile().getAbsolutePath());
126.                 //为 bufferedWrite 分配空间
127.                 bufferedWriter = new Buff-
    eredWriter(fileWriter);
128.                 //保存进去
129.                 bufferedWriter.write(this.jTex-
    tArea.getText());
130.
131.             } catch (Exception e2) {
132.                 e2.printStackTrace();
133.             }finally{
134.                 //关闭文件
135.                 try {
136.                     bufferedWriter.close();
137.                     fileWriter.close();
138.                 } catch (Exception e3) {
139.                     e3.printStackTrace();
140.                 }
141.             }
142.         }
143.     }
144. }
```

(2) 运行截图

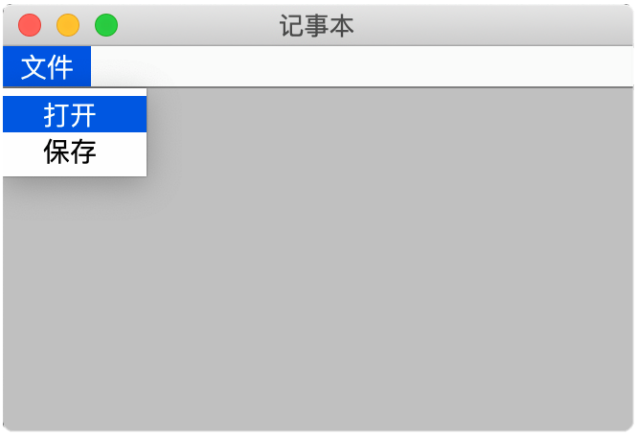
a) 从键盘输入文本

b) “保存”菜单

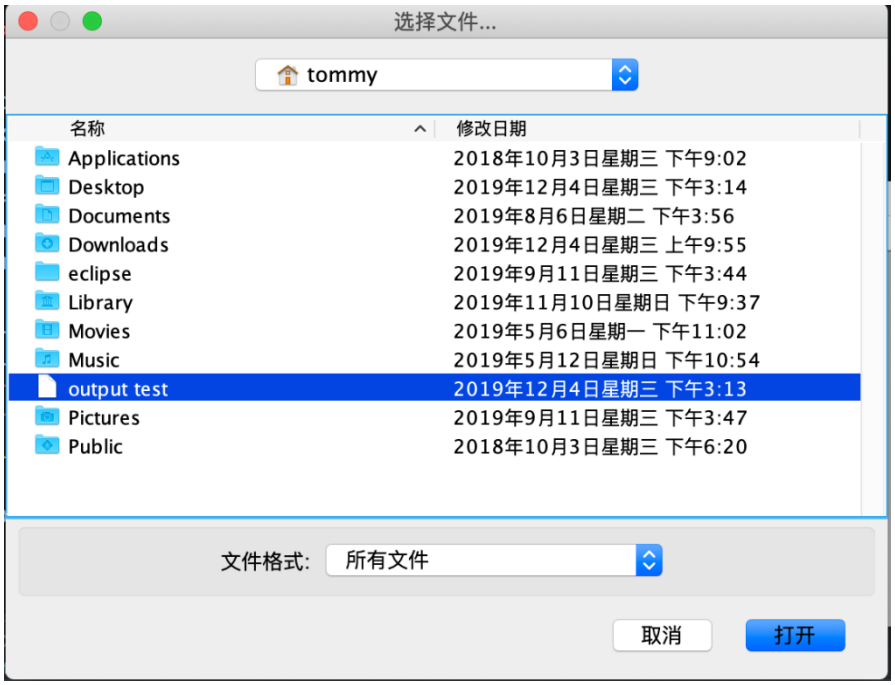
c) 另存为文件



d) “打开”菜单



e) 选择文件打开



九、 总结及心得体会：

本实验目的是让学生理解 Java 异常处理、多线程、I/O、网络以及 GUI 等高级编程技术。

通过本实验，我深入体会到了面向对象程序设计的基本思想，掌握了简单的 Java 异常处理方法，事件了 Java 简单的多线程、I/O、网络及 GUI 高级编程技术。提高了我对问题的分析能力以及编码能力，受益良多。

十、 对本实验过程及方法、手段的改进建议：

实验内容存在重复的题目，且部分题目的时效性不强。可以采用一些新颖的题目背景，激发学生的编程兴趣、提高实验能力。

报告评分：

指导教师签字：