

苏州大学实验报告

院、系	计算机学院	年级专业	19 计科图灵	姓名	张昊	学号	1927405160
课程名称	Java 程序设计					成绩	
指导教师	孔芳	同组实验者	无	实验日期	2021 年 6 月 1 日		

实验名称 上机综合 3

一、实验目的

- (1) 熟练进行 GUI 编程，包括布局管理、事件处理和常用容器和控件
- (2) 联合异常类
- (3) 联合文件类
- (4) 联合集合类

二、实验内容

新建 Java 项目，命名为 `guiprj`，在项目中新建一个名为 `symexample` 包，在该包内新建下面的各个类。设计主窗口，假设类名为 `FrameMain`，含有菜单栏（`JMenuBar`, `JMenu` 和 `JMenuItem`），菜单包括“注册”，“查询”，“退出”，其中：

1. 点击“注册”菜单，弹出一个类名为 `FrameRegister` 窗口，

(1) 请在该窗口用文本框，标签，下拉列表，列表，复选框，单选框和按钮等设计一个学生信息（学号，姓名，性别，城市，爱好等）注册界面；

(2) 在该界面中放置一个“新增”按钮，可以把用户设定的注册信息存到一个集合类中（建议设定一个 `Student` 类，再新建一个 `ArrayList<Student> allstudents` 对象，新增的学生信息先存放到 `allstudents` 中）；

(3) 再放置一个“保存”按钮，点击该按钮，可以把用户此次输入的所有学生的注册信息（建议把 `allstudents` 里的所有学生一个个取出进行存放，不要直接把 `allstudents` 存放到文件）存储到 `guiprj` 项目根目录下的 `student.txt` 文件中（先判断文件是否存在，如果存在，要以新内容可以添加到文件尾部的方式打开文件，即需要支持多次注册）；

(4) 以上功能实现可以选择适合的任意文件流进行数据的存储，注意必要的异常处理。

2. 点击“查询”菜单，弹出一个类名为 `FrameBrowse` 窗口，

(1) 界面类似“注册”窗口。

(2) 打开该窗口时，它自动读取 `student.txt` 里的内容，把里面的每个学生读取出来放到一个 `HashMap` 结构中，里面关键字 `key` 是学号，其它信息想办法存储到它的 `value` 项。默认把第一个学生的信息在界面中相应的控件里显示出来。

(3) 放置一个名为“next”的按钮，点击它，可以提取 `HashMap` 并在界面里展现下一个学生的信息，放置一个名为“previous”的按钮，点击它，可以提取和展现前一个学生的信息。

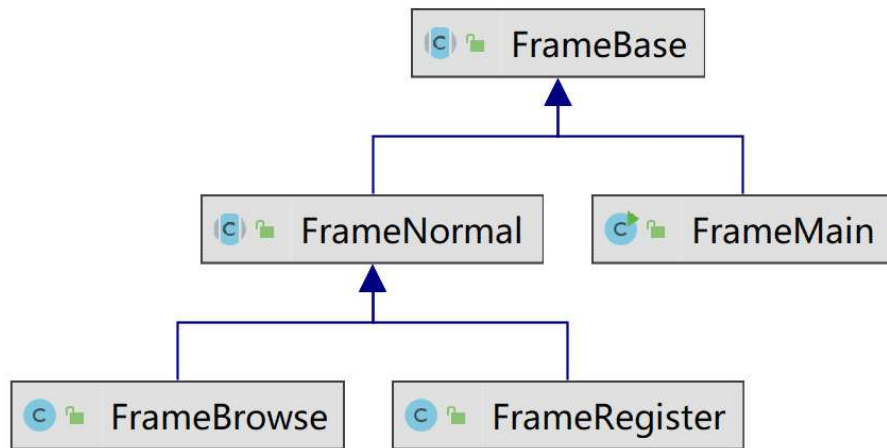
3. 点击“退出”菜单，点击后弹出“是否真要退出？”的对话框，点击“否”“不退出，点击”确定“退出整个系统。

三、实验过程

1. 类的设计

本程序需三个窗口：`FrameMain`, `FrameRegister`, `FrameBrowse`，其中 `FrameBrowse` 和 `FrameRegister` 的界面类似。故设计学生信息显示基类 `FrameNormal`，在其中实现各组件（主要为显示学生信息）的显示，使 `FrameBrowse` 和 `FrameRegister` 类继承自 `FrameNormal`，业

务逻辑的实现交给子类。更进一步，为保证全局 UI 的一致性，设计所有窗口的公共抽象基类 `FrameBase`，为所有窗口提供一致的显示逻辑和共享的常量定义。上述类的继承关系如下图所示：



此外，为在查询中表示学生信息，设计学生类 `Student`；为完成文件的读取与写入，设计文件管理类 `FileManager` 类。

2. 公共窗口抽象基类 `FrameBase` 提供的显示逻辑

在类中提供了统一的显示逻辑：构造函数接受三个参数，分别为窗口标题，窗口大小（类的静态常量提供了两种固定的窗口大小可供选择），并负责初始化实例变量。当打开一个窗口时需要手动调用 `start` 方法使窗口显示，此时会调用类中的 `onStart` 方法和 `refresh` 方法，其中 `onStart` 方法为预留给子类覆盖的方法，用于实现窗口的显示逻辑。当一个窗口被关闭时会自动触发 `finish` 方法，此时会调用 `onFinish` 方法，并根据方法返回值确定是否关闭窗口。当一个窗口需要刷新显示区域时需调用 `refresh` 方法，此时会调用 `onRefresh` 方法。`start`、`finish`、`refresh` 方法设计为 `public final` 方法，子类不可覆盖，`protected` 的 `onStart`、`onFinish`、`onRefresh` 方法只是简单地留空，便于子类覆盖。

`FrameBase` 类被设计为抽象的，但不存在抽象方法，子类可以有选择性地实现 `onStart`、`onFinish`、`onRefresh` 方法以完成显示逻辑。关键代码如下：

```
public final void start() {
    this.onStart();
    this.refresh();
}

public final void finish() {
    if (this.onFinish()) {
        this.dispose();
    }
}

public final void refresh() {
    this.onRefresh();
    this.setVisible(true);
}
```

```

}
// 交互逻辑子类覆盖方法
protected void onStart() {
    this.content.setLayout(new BorderLayout());
}
protected boolean onFinish() {
    return true;
}
protected void onRefresh() {}

```

3. 主窗口设计

主窗口类包含学生注册类和学生查询类的引用，用于表示两个窗口的打开情况，以防一次性打开多个窗口导致的不一致性问题，其管理逻辑详见菜单栏实现中各按钮的事件监听器。菜单栏实现如下：

```

private void initMenu() {
    this.setJMenuBar(menuBar);
    this.addMenu("注册", new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            if (registerWindow == null) {
                registerWindow = new FrameRegister();
                registerWindow.start();
            } else if (!registerWindow.isVisible()) {
                registerWindow.refresh();
            } else if (registerWindow.getState() == Frame.ICONIFIED) {
                registerWindow.setExtendedState(Frame.NORMAL);
                registerWindow.requestFocus();
            } else if (!registerWindow.isFocused()) {
                registerWindow.requestFocus();
            }
        }
    });
    this.addMenu("查询", new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            if (browseWindow == null) {
                browseWindow = new FrameBrowse();
                browseWindow.start();
            } else if (!browseWindow.isVisible()) {
                browseWindow.refresh();
            } else if (browseWindow.getState() == Frame.ICONIFIED) {
                browseWindow.setExtendedState(Frame.NORMAL);
                browseWindow.requestFocus();
            } else if (!browseWindow.isFocused()) {

```

```

        browseWindow.requestFocus();
    }
}
});
this.addMenu("退出", event -> this.finish());
}

// 添加单个菜单
private void addMenu(String menuName, ActionListener
actionListener) {
    JMenu menu = new JMenu(menuName);
    menu.setFont(FONT_NORMAL);
    this.menuBar.add(menu);
    JMenuItem item = new JMenuItem(menuName);
    item.setFont(FONT_PLAIN);
    menu.add(item);
    item.addActionListener(actionListener);
}

```

点击“退出”菜单，点击后弹出“是否真要退出？”的对话框，点击“否“不退出，点击”确定“退出整个系统。

得益于抽象基类 FrameBase 的设计，本类中只需重写 onFinish 方法即可实现：

```

@Override
public boolean onFinish() {
    this.closeAllWindows(); // 关闭所有窗口
    return JOptionPane.showConfirmDialog(this,
        "是否真要退出？",
        "退出系统",
        JOptionPane.YES_NO_OPTION) == JOptionPane.OK_OPTION;
}

```

4. 学生信息显示基类 FrameNormal 设计

在此类中，使用 JTextField 实现学生学号、姓名、城市，使用 JRadioButton 事项学生性别选择，使用 JComboBox 实现所在学院。以上组件采用 BoxLayout 布局，通过控制是否可编辑以及 setText 等来控制其是展示信息还是接受信息。由于爱好信息在不同窗口显示逻辑不同，故为其提供显示区域 favoriteContent，以及 clearFavorite，accessFavorite，putFavorite，setFavoriteToShow 方法，前三者作为抽象方法要求子类根据爱好的显示方法不同来实现，后者供子类选择性实现。

由于每个窗口都有两个按钮，故为子类提供两个方法 onClickLeftButton 和 onClickRightButton，用以实现点击两按钮时的动作。

此外，本类通过两个方法 putStudentToView，getStudentFromView，分别实现向窗口展示学生信息以及从窗口获取学生信息。

该类的实现较为复杂，详细代码请参见附件。

5. 学生注册窗口类设计

基于 `FrameNormal` 类，我们可以简单实现各方法即可实现一个学生注册窗口的显示逻辑。类中使用 `ArrayList` 作为数据的缓冲区，设置窗口各组件为可编辑状态，学生爱好采用 `JCheckBox`（多选框）实现。

左按钮设计为“新增”，把用户设定的注册信息存到 `ArrayList students` 中：

```
@Override
protected void onClickLeftButton() {
    Student student = this.getStudentFromView(false);
    if (student != null) {
        this.students.add(student);
        this.clear();
        this.setBottomLabelText(String.format("已添加 %d 名学生",
this.students.size()));
    }
}
```

右按钮设计为“保存”，将 `ArrayList` 中的数据保存至文本文件。首先确认一下输入区域是否存在合法的但还未“新增”的学生信息，如有，先加到缓冲区中。之后再通过文件管理器类将学生数据保存至文本文件。最后会弹窗提示是否成功。实现如下：

```
@Override
protected void onClickRightButton() {
    this.checkNotAddedStudent();
    if (this.students.size() == 0) { // 没有添加学生信息
        JOptionPane.showMessageDialog(this,
            "还没有添加学生信息",
            "注册失败",
            JOptionPane.ERROR_MESSAGE);
        return;
    }
    boolean success = true;
    try {
        FileManager manager = new FileManager(Student.SAVED_FILE_NAME);
        for (Student student : this.students) {
            manager.writeStudent(student);
        }
    } catch (IOException e) { // 出现错误
        e.printStackTrace();
        success = false;
        JOptionPane.showMessageDialog(this,
            String.format("%d 名学生注册时出现错误\n%s",
                this.students.size(), e.getMessage()),
            "注册失败",
            JOptionPane.INFORMATION_MESSAGE);
    }
    if (success) { // 注册成功
        JOptionPane.showMessageDialog(this,
```

```

        String.format("成功注册 %d 名学生", this.students.size()),
        "注册成功",
        JOptionPane.INFORMATION_MESSAGE);
        this.setBottomLabelText("已添加 0 名学生");
        this.students.clear();
    }
}

// 检查显示区域还有未新增的学生信息，如有，询问是否新增
private void checkNotAddedStudent() {
    Student tempStudent = this.getStudentFromView(true);
    if (tempStudent != null) {
        if (JOptionPane.showConfirmDialog(this,
            "显示区域还有未新增的学生信息，是否新增？",
            "未新增的学生",
            JOptionPane.YES_NO_OPTION) == JOptionPane.OK_OPTION) {
            this.students.add(tempStudent);
            this.setBottomLabelText(String.format("已添加 %d 名学生",
                this.students.size()));
        }
        this.clear();
    }
}
}

```

此外，在本窗口关闭时还会检查是否有没保存的数据，通过重写 onFinish 实现：

```

@Override
protected boolean onFinish() {
    this.checkNotAddedStudent();
    if (!students.isEmpty()) {
        if (JOptionPane.showConfirmDialog(this,
            String.format("还有 %d 个学生信息未保存，是否在关闭前保存？",
                this.students.size()),
            "未保存的学生",
            JOptionPane.YES_NO_OPTION) == JOptionPane.OK_OPTION) {
            this.onClickRightButton();
        } else {
            this.students.clear();
            this.setBottomLabelText("已添加 0 名学生");
        }
    }
    return true;
}
}

```

6. 学生查询（浏览）窗口类实现

类中使用 HashMap 来保存学生信息，key 为学生学号（字符串），value 为学生类 Student

对象。并且使用 List 来保存学生的学号索引（便于顺序遍历），使用三个整型变量 prev, index, next 在遍历时标明前一个，当前，后一个学生索引下标（keys）。设置窗口各组件为不可编辑状态，学生爱好采用 JLabel 实现。

左按钮为“previous”，获取并显示前一个学生，若不存在（返回 null）提示信息：

```
@Override
protected void onClickLeftButton() {
    Student student = getPreviousStudent();
    this.showStudentIndex();
    if (student == null) {
        JOptionPane.showMessageDialog(this,
            (this.keys.size() == 0) ? "没有可显示的学生":"已经是第一个学生",
            "没有更多学生",
            JOptionPane.ERROR_MESSAGE);
        return;
    }
    this.putStudentToView(student);
}
private Student getPreviousStudent() {
    if (this.prev <= -1) {
        return null;
    }
    Student student = this.students.get(this.keys.get(this.prev));
    this.next--; this.index--; this.prev--;
    return student;
}
```

右按钮为“next”，获取并显示下一个学生，若不存在（返回 null）提示信息：

```
@Override
protected void onClickRightButton() {
    Student student = getNextStudent();
    this.showStudentIndex();
    if (student == null) {
        String msg = "已经是最后一个学生";
        if (this.keys.size() == 0) {
            msg = "没有可显示的学生";
        }
        JOptionPane.showMessageDialog(this, msg, "没有更多学生",
            JOptionPane.ERROR_MESSAGE);
        return;
    }
    this.putStudentToView(student);
}
private Student getNextStudent() {
    if (this.next >= this.students.size())
        return null;
}
```

```

        Student student = this.students.get(this.keys.get(this.next));
        this.next++;
        this.index++;
        this.prev++;
        return student;
    }

```

7. 文本文件保存与读取，Student 类的设计

FileManager 类构造函数会检查传入的文件是否存在，若不存在则创建。使用 BufferedWriter/BufferedReader 来追加写入/读取文件，其中，读取文件会检查文件中是否存在重复的学生学号，若存在则抛出异常。具体实现详见附件。

Student 类设计为保存学生信息。其中提供了静态方法 parseStudent，实现从 toString 转换为的字符串重新解析为 Student 对象：

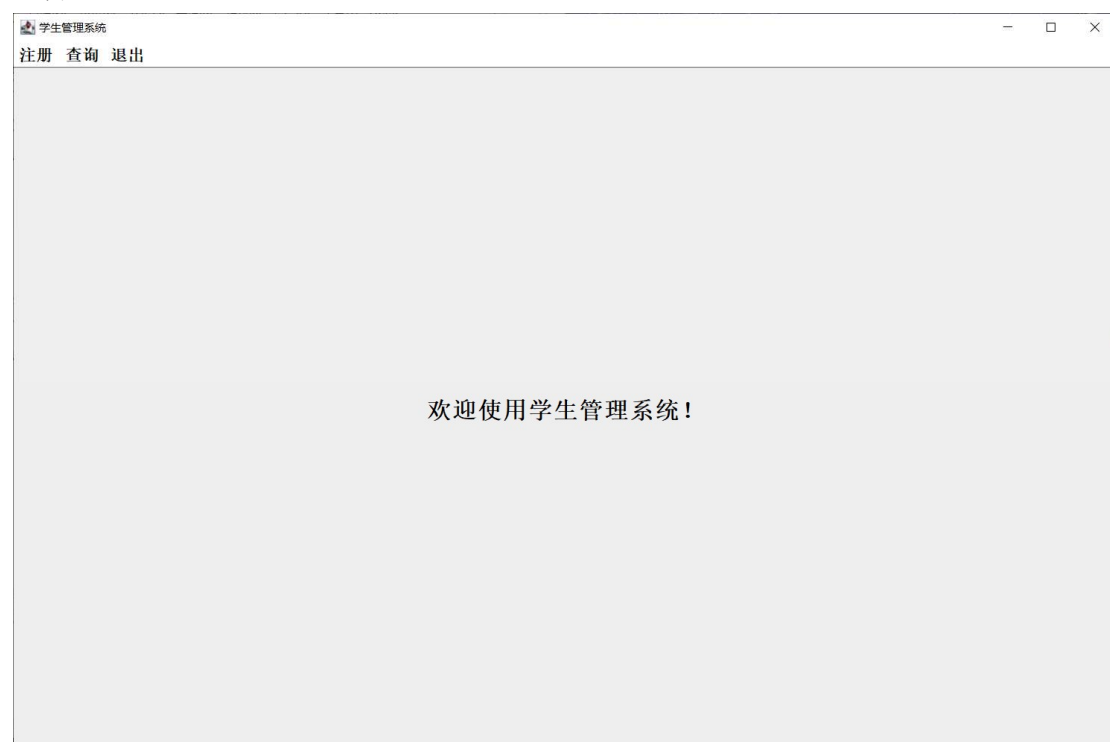
```

public static Student parseStudent(String string) throws
RuntimeException {
    String[] line = string.trim().split(FIELD_SEPARATOR);
    List<String> favorites = new LinkedList<>();
    if (line.length == 6) {
        favorites = Arrays.asList(line[5].trim().split(FAVORITE_SEPARATOR));
    }
    return new Student(line[0], line[1], line[2], line[3],
        line[4], favorites);
}

```

四、实验结果

主窗口：



注册窗口：

学生注册

— □ ×

当前位置：学生注册

学号 *

姓名 *

城市 *

性别 *

☐ 男

☐ 女

学院 *

爱好

☐ 读书

☐ 听音乐

☐ 跑步

☐ 乒乓球

☐ 羽毛球

☐ 滑雪

新增

保存

已添加 0 名学生

成功保存提示：

注册成功

×



成功注册 1 名学生

确定

存在未新增的学生时的提示：

未新增的学生

×

?

显示区域还有未新增的学生信息，是否新增？

是(Y)

否(N)

学生查询窗口：

学生查询

— □ ×

当前位置：学生查询

学号 *

1927405160

姓名 *

张昊

城市 *

苏州

性别 *

☒ 男

☐ 女

学院 *

计算机科学与技术学院 ▾

爱好

读书

previous

next

当前为第 5 名学生，共 6 条数据

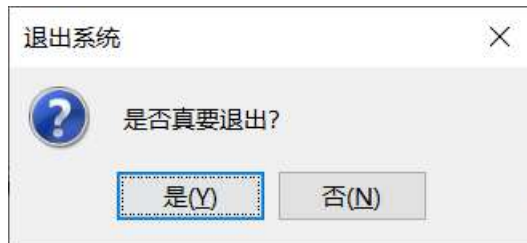
向后遍历至末尾提示：



向前遍历至末尾提示:



主窗口提出询问:



五、实验总结

通过实验,我熟练掌握了 GUI 编程,深刻地认识了布局管理、事件处理和常用容器和控件,并进行了简单的封装以方便二次开发。同时,我也熟悉了 Java 异常处理与 IO,以及各种集合类的使用,如 ArrayList、HashMap 等等。