

Lab 文档

1、工具

本次 lab 在 eclipse 上开发，使用语言是 java。

2、打开方式

(1) 直接使用 jar 包

(2) 执行 **MainFrame** 类中的 **main** 函数，会出现 GUI 界面，此时点击读入，选择读入文件，会出现文件选择器，选择一个有命题集的 **txt** 格式文件，开始绘制。

点击绘制，会画出当前命题的形成树，如果命题不能绘制，则会显示“此命题格式不正确”。绘制键有 **next** 键的功能。

命题可以绘制：直接画出树。

命题不可以绘制：显示文字“此命题格式不正确”。

3、类的结构

本次 lab 的实现的类如下：

- 1、 主类 **MainFrame**，作为主界面
- 2、 **panel** 类，作为绘图界面
- 3、 **Tree**
- 4、 **TreeNode**

MainFrame 实现了按键，菜单栏的功能，并用一个 **scrollpane** 容纳 **panel**，**panel** 依据树的大小来计算，如果树过大会出现滚动条。

Panel 中有绘制树每个节点的方法，并实现 **repaint** 后动态的计算 **panel** 大小。

Tree 实现了树结构，包含构造树的方法和初步检查命题的方法。

TreeNode 作为树的节点，存储命题。

具体每个类的方法如何实现在代码注释中已经说明。

4、实现过程

首先 **MainFrame** 中，读入 **txt** 文件后，会形成一个 **String** 数组。如果没有读入就点击绘制会提示“请读入文件”。

点击绘制进行绘制，此时会先对 **String** 数组中的一个 **string** 调用

`checkComponent` 方法进行初步的检查，如正反括号数目不一致，出现规定外的字符串等情况会直接告诉使用者命题格式有误。

如果通过初步检查后，会构建树，调用 `Tree` 中的 `formationTree` 方法，`formationTree` 的方法实现依据 PDF 中的算法。形成树的过程中，如果有异常，会告诉使用者命题格式有误。

如果成功形成一棵树，将会调用 `repaint` 方法进行绘制。绘制方法在 `panel` 类中，主要是 `paintNode` 方法，这个方法用递归的方式绘制每个节点的字符串，并与其父节点连线。

点击绘制进行下一个命题的操作。如果到了最后一个 `string` 将会提示没有下一个命题。