软件体系结构作业 20

姓　名：洪伟鑫

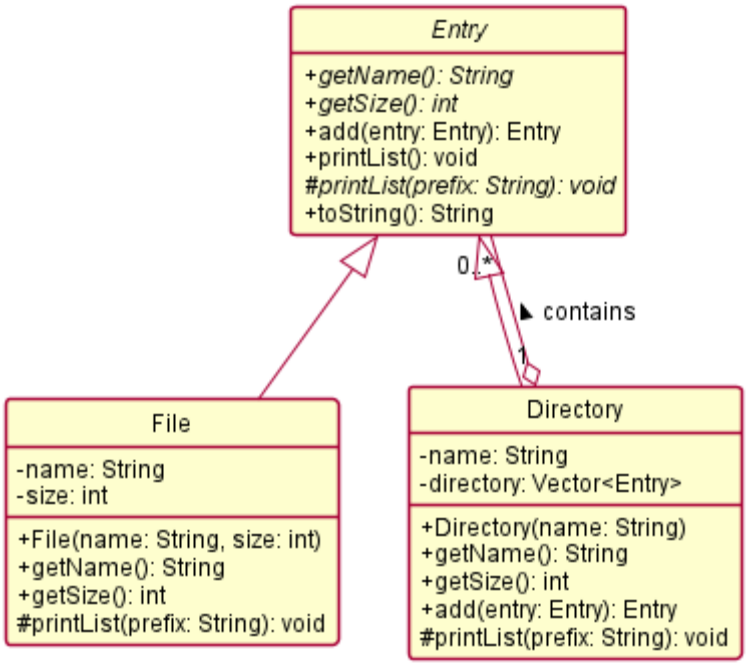
专　业：软件工程

年　级：2022 级

学　号：37220222203612

2025 年 5 月 17 日

1、用 GUI 改写本例并访问你的计算机某个子目录。



- 组合设计模式用来构建一个文件系统的层级模型：核心在于一个抽象的 Entry 类，它为文件（ File 类，代表不能包含其他条目的叶子节点）和目录（ Directory 类，代表可以包含其他文件或子目录的组合节点）定义了统一的操作接口，如获取名称、大小和打印列表。
- Directory 类内部持有一个 Entry 对象的集合，从而能够递归地组织这些条目，形成一个树状结构。
- 这种设计使得客户端代码可以一致地处理单个文件和复杂的目录结构，无需关心它们的具体类型，极大地简化了对层级数据的操作。

```
public DirectoryBrowser() {
    setTitle("目录浏览器");
    setSize(600, 400);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    // 为BorderLayout添加水平和垂直间隙
    setLayout(new BorderLayout(5, 5));

    // 面板用于放置按钮和标签
    // 为FlowLayout添加水平和垂直间隙
    JPanel topPanel = new JPanel(new FlowLayout(FlowLayout.LEFT, 10, 5));
    browseButton = new JButton("选择目录...");
    selectedPathLabel = new JLabel("尚未选择目录");

    // --- 添加颜色 ---
    topPanel.setBackground(new Color(230, 230, 250)); // 淡紫色背景
    selectedPathLabel.setForeground(new Color(50, 50, 100)); // 深蓝色文本，以搭配背景

    topPanel.add(browseButton);
    topPanel.add(selectedPathLabel);
    // 为topPanel添加内边距
    topPanel.setBorder(BorderFactory.createEmptyBorder(5, 5, 0, 5));

    add(topPanel, BorderLayout.NORTH);

    // 初始化空的JTree
    DefaultMutableTreeNode rootNode = new DefaultMutableTreeNode("请选择一个目录");
    directoryTree = new JTree(rootNode);
    treeScrollPane = new JScrollPane(directoryTree);

    // --- 添加颜色 ---
    treeScrollPane.getViewport().setBackground(new Color(245, 248, 255)); // 非常浅的蓝色背景
    directoryTree.setBackground(new Color(250, 250, 255)); // 给JTree本身也设置一个相近的颜色

    // 为treeScrollPane添加内边距
    treeScrollPane.setBorder(BorderFactory.createEmptyBorder(0, 10, 10, 10)); // 增加了左右和下边距

    add(treeScrollPane, BorderLayout.CENTER);

    browseButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            JFileChooser fileChooser = new JFileChooser();
            fileChooser.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);
            fileChooser.setDialogTitle("选择一个目录");

            int returnValue = fileChooser.showOpenDialog(DirectoryBrowser.this);
            if (returnValue == JFileChooser.APPROVE_OPTION) {
                File selectedFile = fileChooser.getSelectedFile();
                selectedPathLabel.setText("当前目录: " + selectedFile.getAbsolutePath());
                populateTree(selectedFile);
            }
        }
    });

    setLocationRelativeTo(null); // 居中显示
}
```

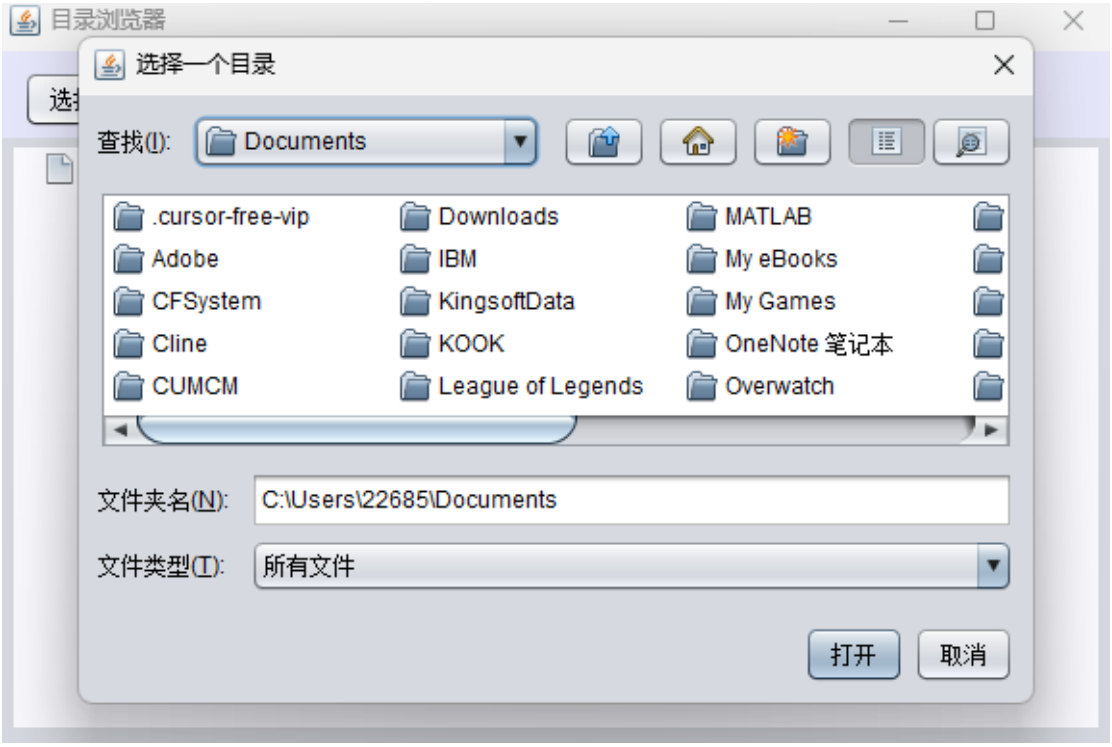
- DirectoryBrowser 类的构造函数，它负责初始化 GUI 的目录浏览器窗口
- 首先，它设置了窗口的标题、大小和关闭行为，并采用带有间隙的 BorderLayout 布局；接着，创建了一个位于顶部的面板（ topPanel ），该面板使用带间隙的 FlowLayout 并包含一个“选择目录...”按钮和一个用于显示所选路径的标签；
- 然后，它初始化了一个 JTree 组件用于显示目录结构，并将其放入一个带浅蓝色背景和内边距的滚动面板（ treeScrollPane ）中，分别将顶部面板和滚动面板添加到窗口的北部和中部；最后，为“选择目录...”按钮添加了事件监听器，使得用户点击按钮后能通过文件选择器选取一个目录，并在选中后更新路径标签和调用 populateTree 方法来展示目录内容，同时窗口会被居中显示。

```java
private void populateTree(File rootDirectory) {
    DefaultMutableTreeNode rootNode = new DefaultMutableTreeNode(new FileNode(rootDirectory));
    createTreeNodes(rootDirectory, rootNode);
    DefaultTreeModel treeModel = new DefaultTreeModel(rootNode);
    directoryTree.setModel(treeModel);
}

private void createTreeNodes(File currentFile, DefaultMutableTreeNode parentNode) {
    if (currentFile.isDirectory()) {
        File[] files = currentFile.listFiles();
        if (files != null) {
            for (File file : files) {
                DefaultMutableTreeNode childNode = new DefaultMutableTreeNode(new FileNode(file));
                parentNode.add(childNode);
                if (file.isDirectory()) {
                    createTreeNodes(file, childNode);
                }
            }
        }
    }
}

// 辅助类，用于在JTree中更好地显示文件名
private static class FileNode {
    private File file;

    public FileNode(File file) {
        this.file = file;
    }

    public File getFile() {
        return file;
    }

    @Override
    public String toString() {
        return file.getName().isEmpty() ? file.getAbsolutePath() : file.getName();
    }
}
```

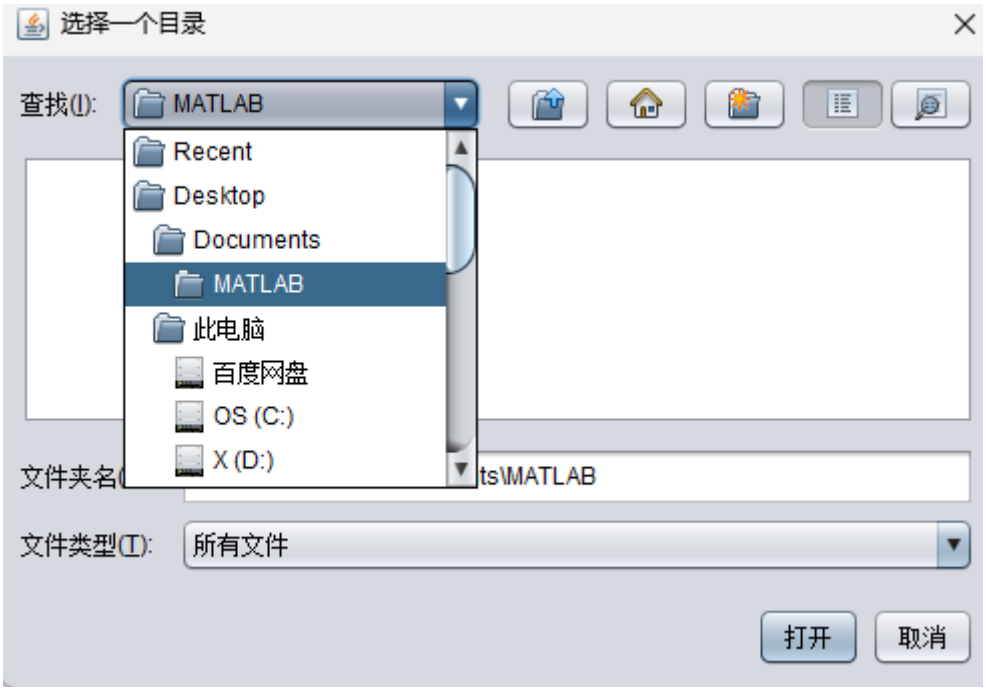
- populateTree 方法，它接收一个根目录，初始化树的根节点并调用 createTreeNodes 来构建子节点，最后将生成的树模型设置给 directoryTree；
- createTreeNodes 方法，它递归地遍历指定目录下的所有文件和子目录，为每一个条目创建一个 DefaultMutableTreeNode （其用户对象是 FileNode 实例），并将其添加到父节点，如果条目是目录则继续递归；以及静态内部类 FileNode，它包装了一个 File 对象，并重写 toString 方法以在树中正确显示文件名或路径。

运行结果：

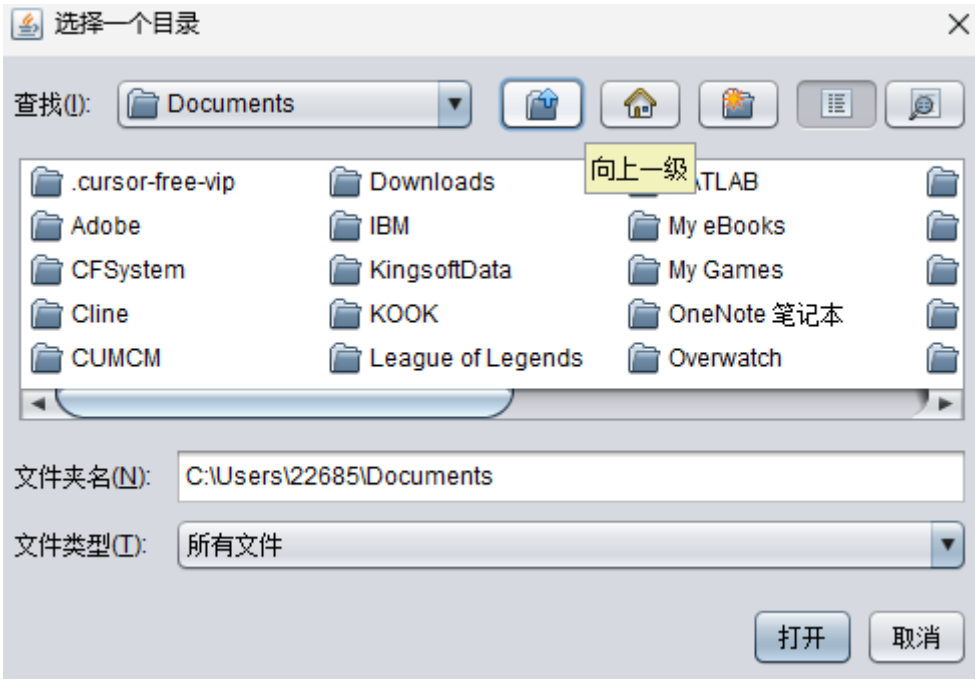
GUI 主界面



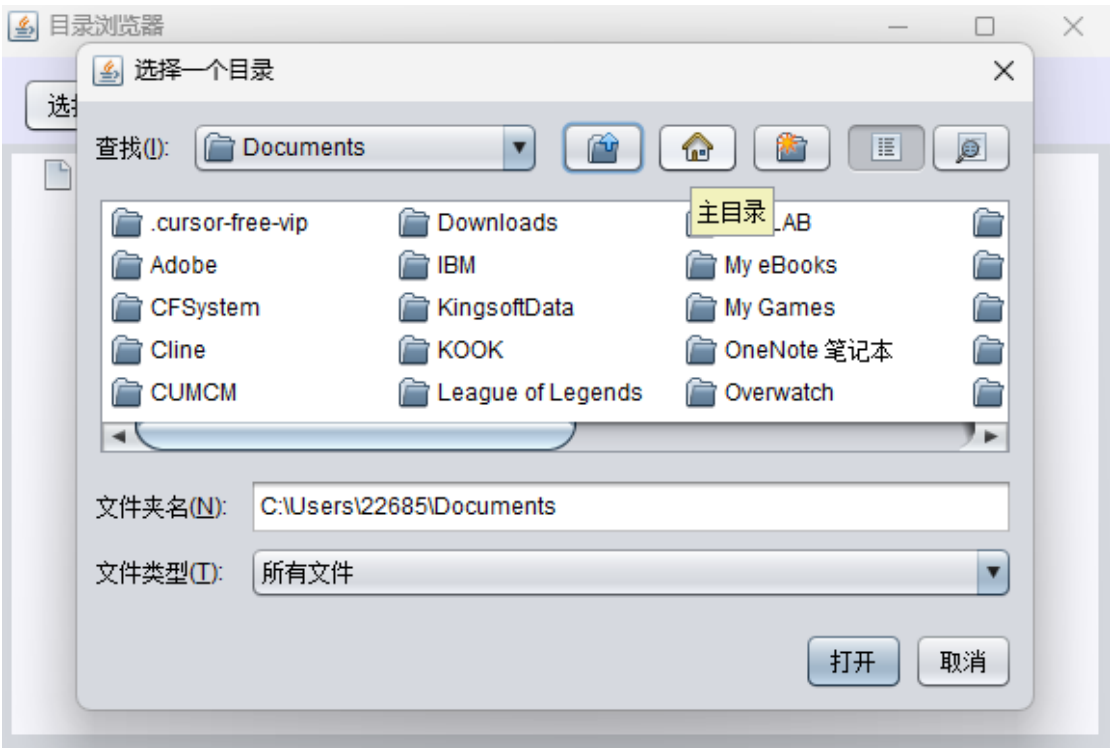
可以选择目录



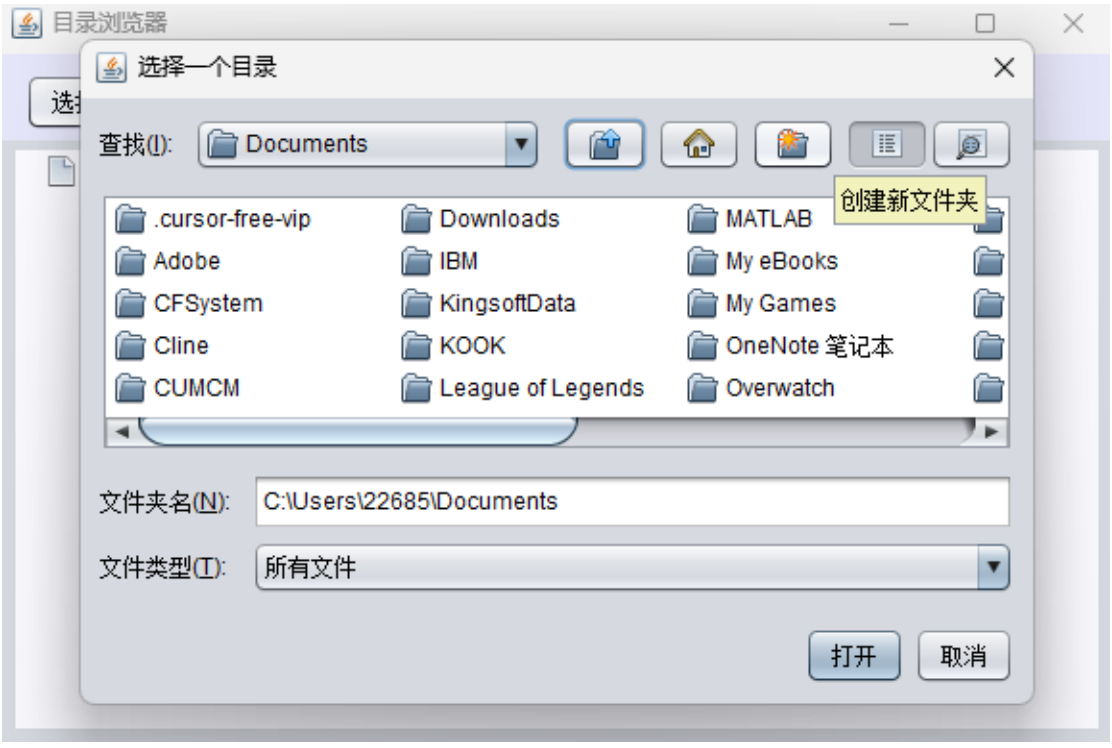
可以查找目录



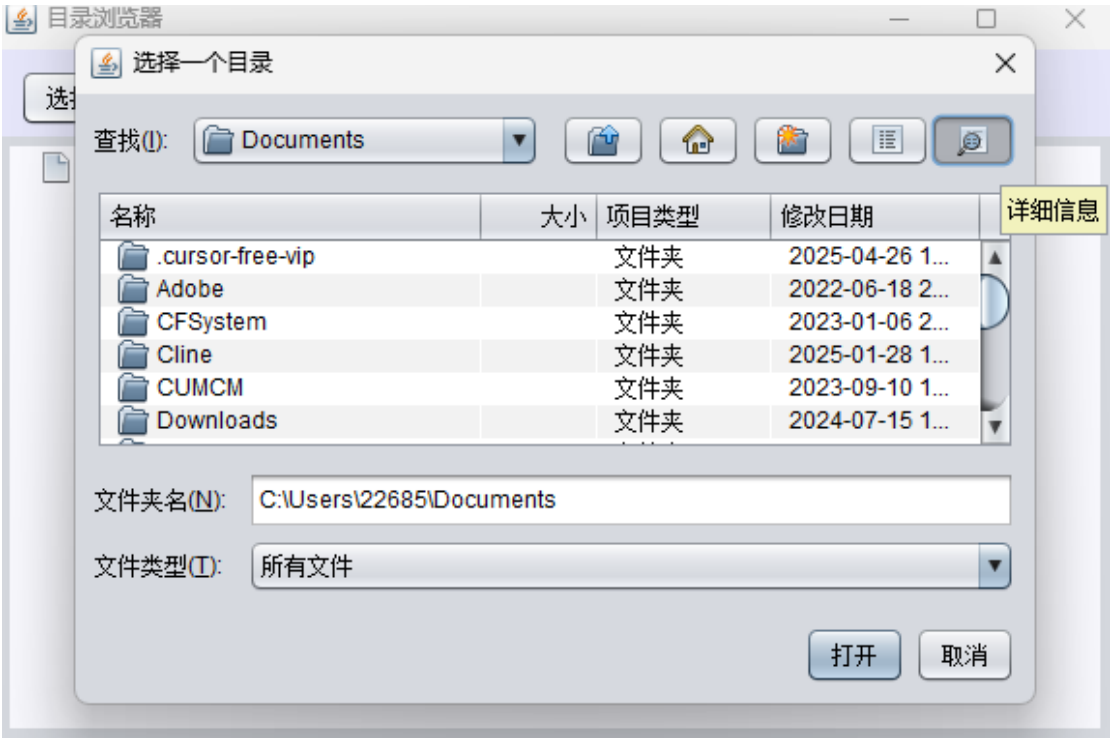
可以向上一级



可以回到主目录



可以创建新文件夹



可以查看详细信息

以打开体系结构目录为例，可以正常使用