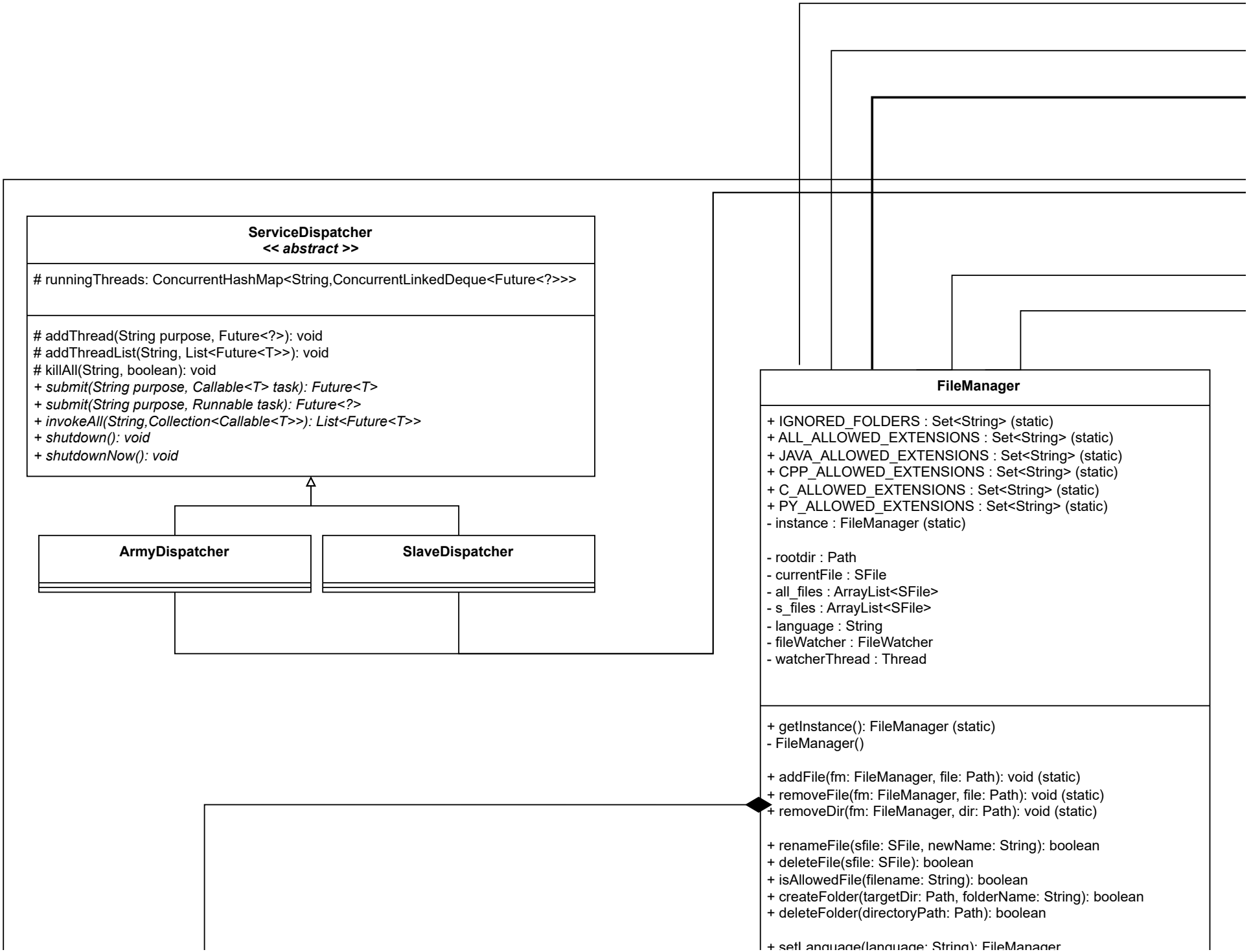
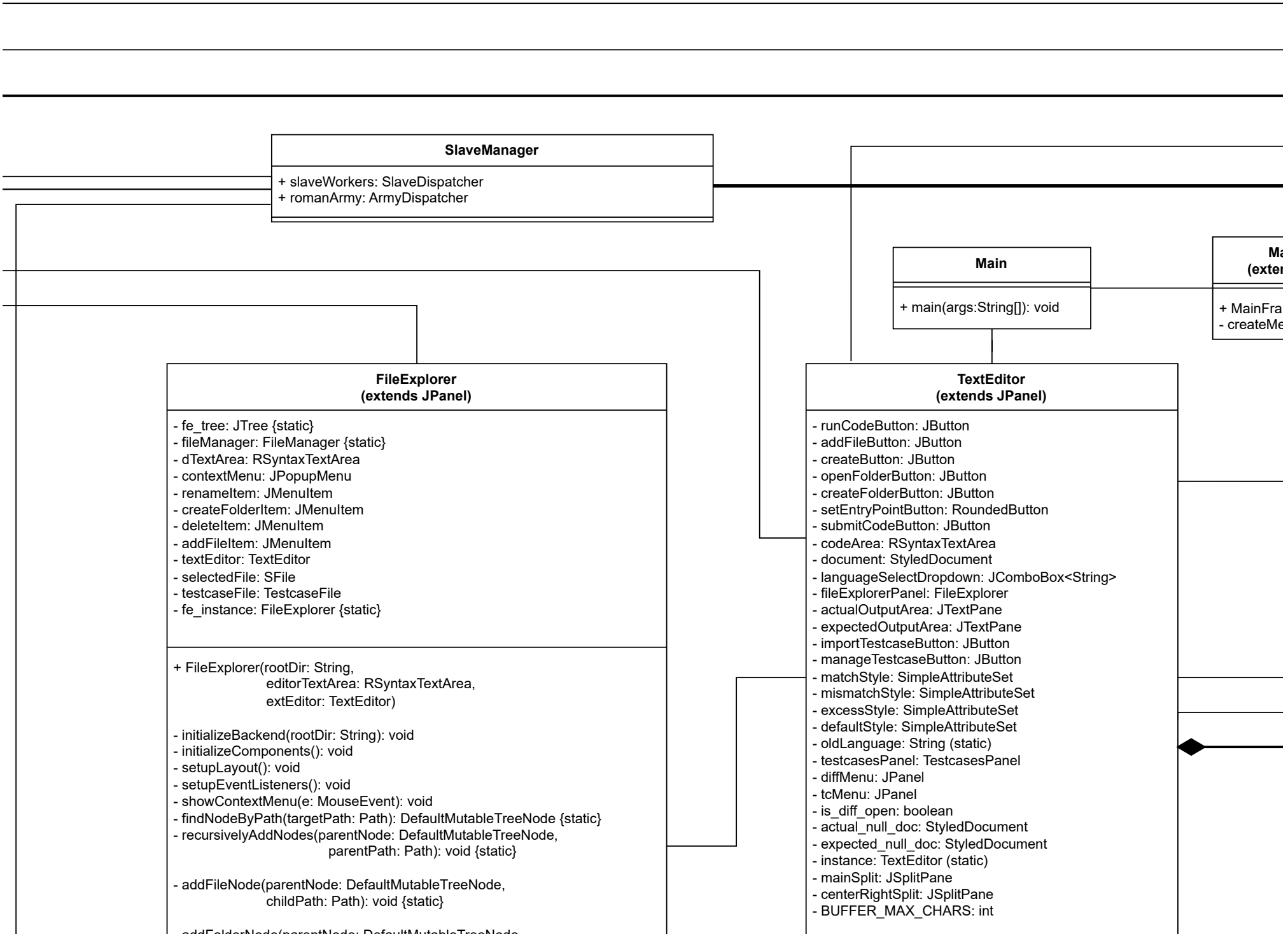
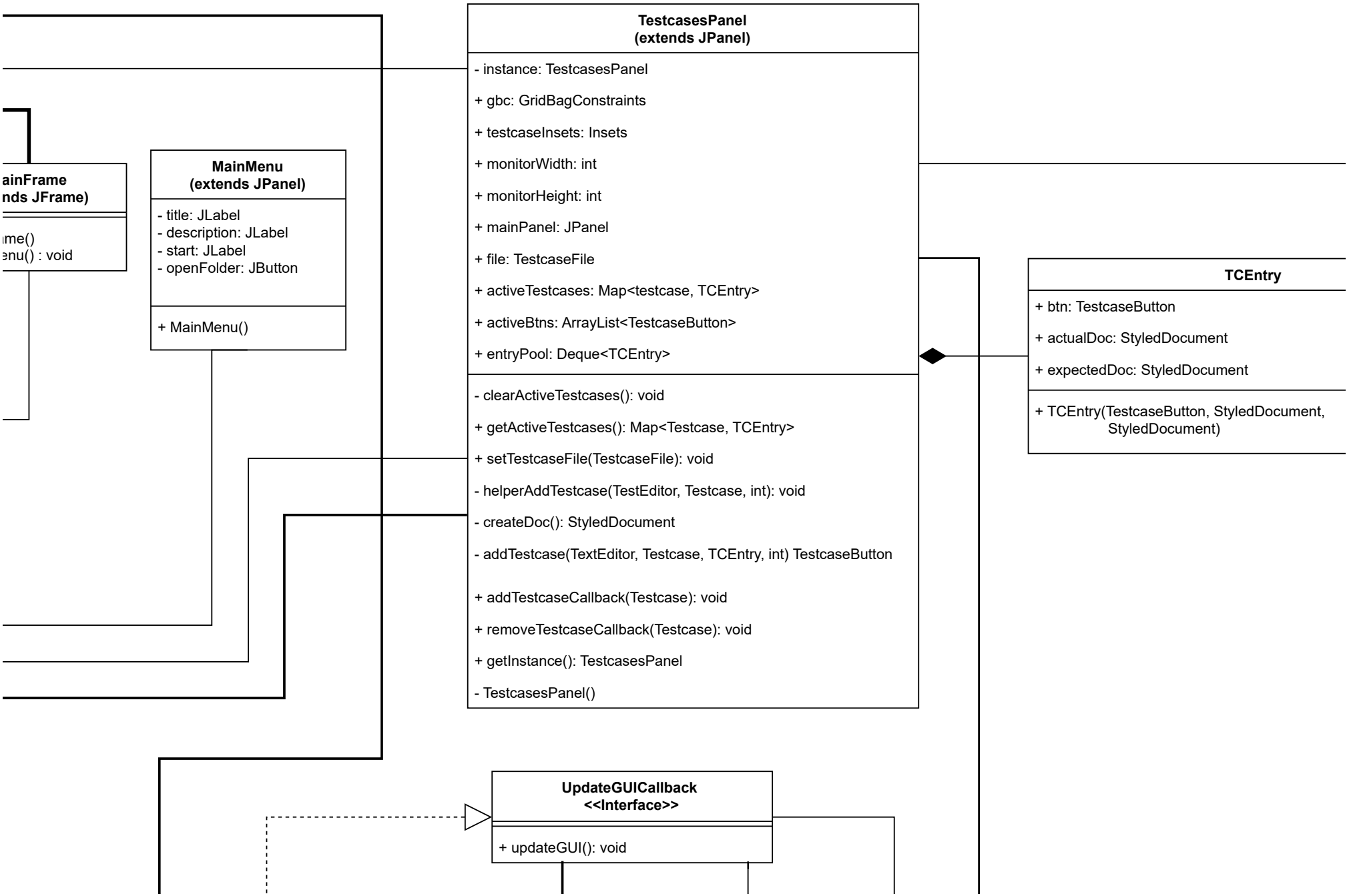
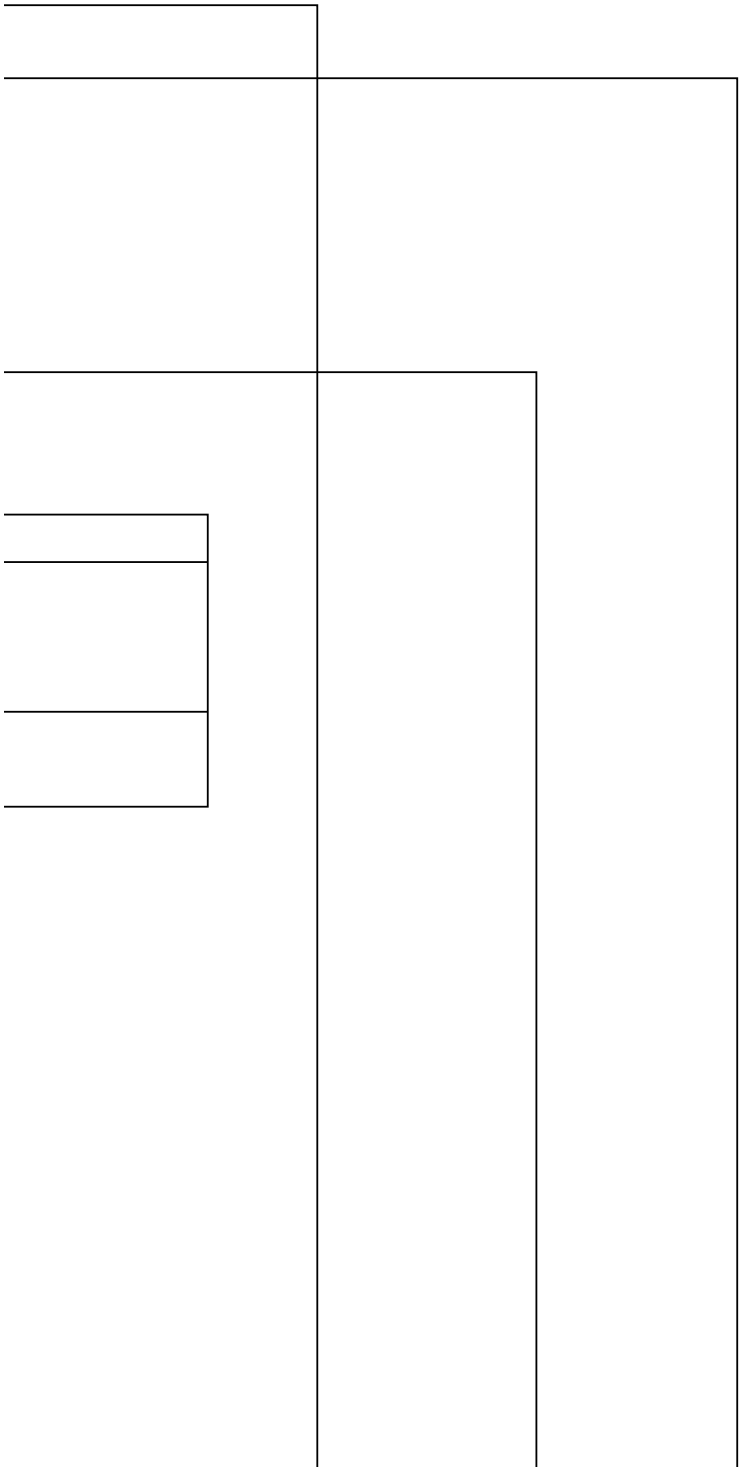
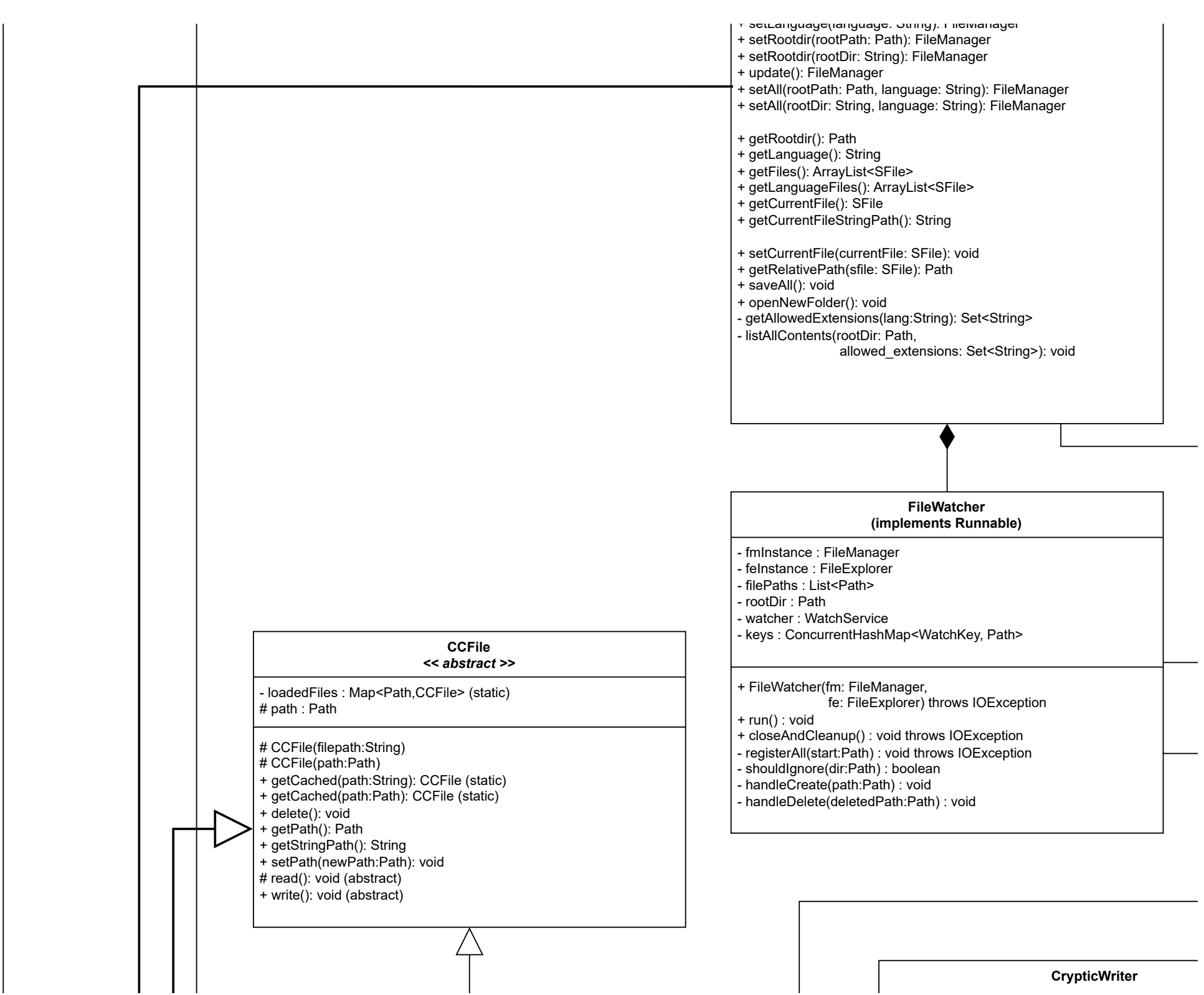
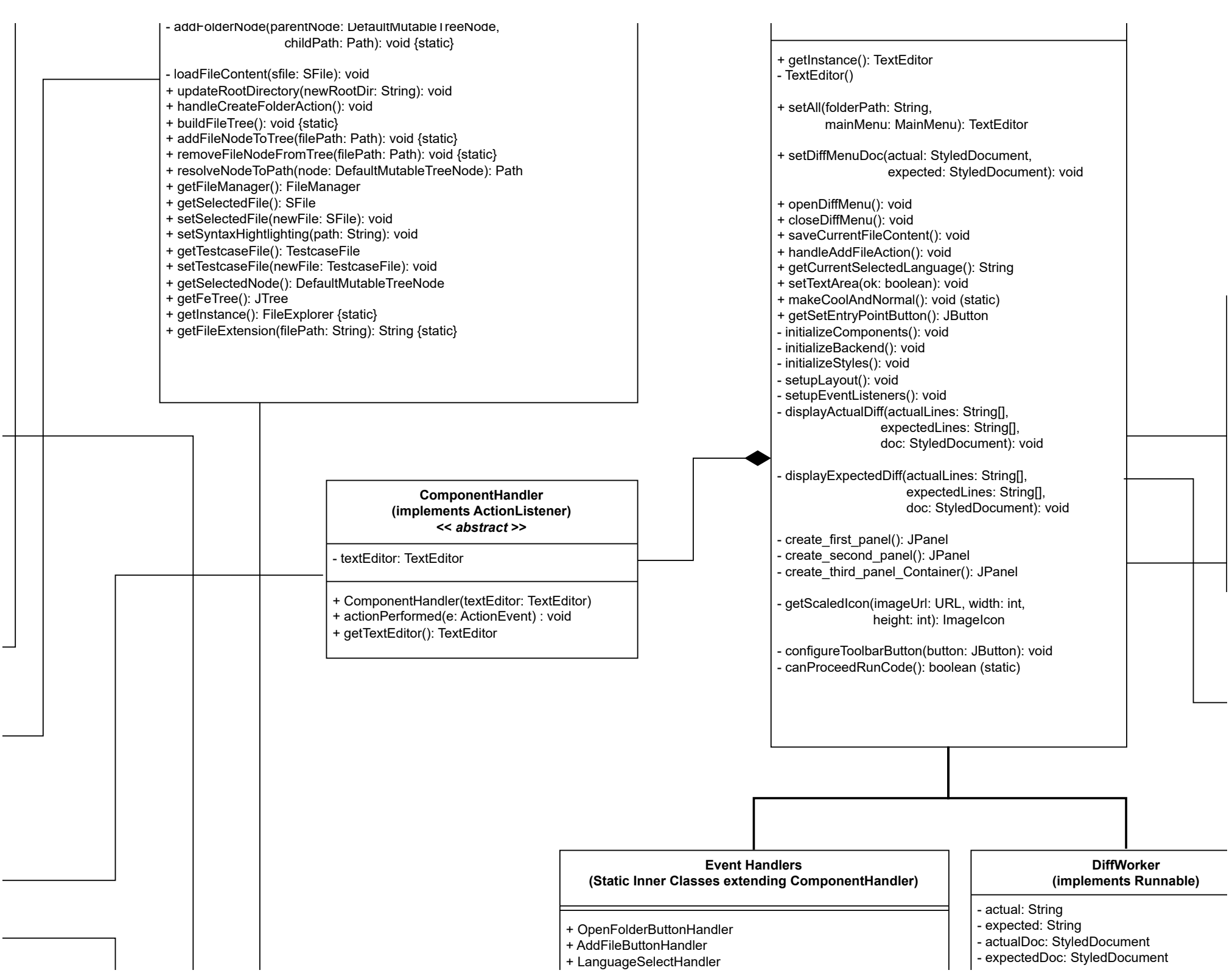

ServiceDispatcher
<< abstract >>
runningThreads: ConcurrentHashMap<String,ConcurrentLinkedDeque<Future<?>>>
addThread(String purpose, Future<?>): void
addThreadList(String, List<Future<T>>): void
killAll(String, boolean): void
+ submit(String purpose, Callable<T> task): Future<T>
+ submit(String purpose, Runnable task): Future<?>
+ invokeAll(String,Collection<Callable<T>>): List<Future<T>>
+ shutdown(): void
+ shutdownNow(): void
ArmyDispatcher
SlaveDispatcher
FileManager
+ IGNORED_FOLDERS : Set<String> (static)
+ ALL_ALLOWED_EXTENSIONS : Set<String> (static)
+ JAVA_ALLOWED_EXTENSIONS : Set<String> (static)
+ CPP_ALLOWED_EXTENSIONS : Set<String> (static)
+ C_ALLOWED_EXTENSIONS : Set<String> (static)
+ PY_ALLOWED_EXTENSIONS : Set<String> (static)
- instance : FileManager (static)
- rootdir : Path
- currentFile : SFile
- all_files : ArrayList<SFile>
- s_files : ArrayList<SFile>
- language : String
- fileWatcher : FileWatcher
- watcherThread : Thread
+ getInstance(): FileManager (static)
- FileManager()
+ addFile(fm: FileManager, file: Path): void (static)
+ removeFile(fm: FileManager, file: Path): void (static)
+ removeDir(fm: FileManager, dir: Path): void (static)
+ renameFile(sfile: SFile, newName: String): boolean
+ deleteFile(sfile: SFile): boolean
+ isAllowedFile(filename: String): boolean
+ createFolder(targetDir: Path, folderName: String): boolean
+ deleteFolder(directoryPath: Path): boolean

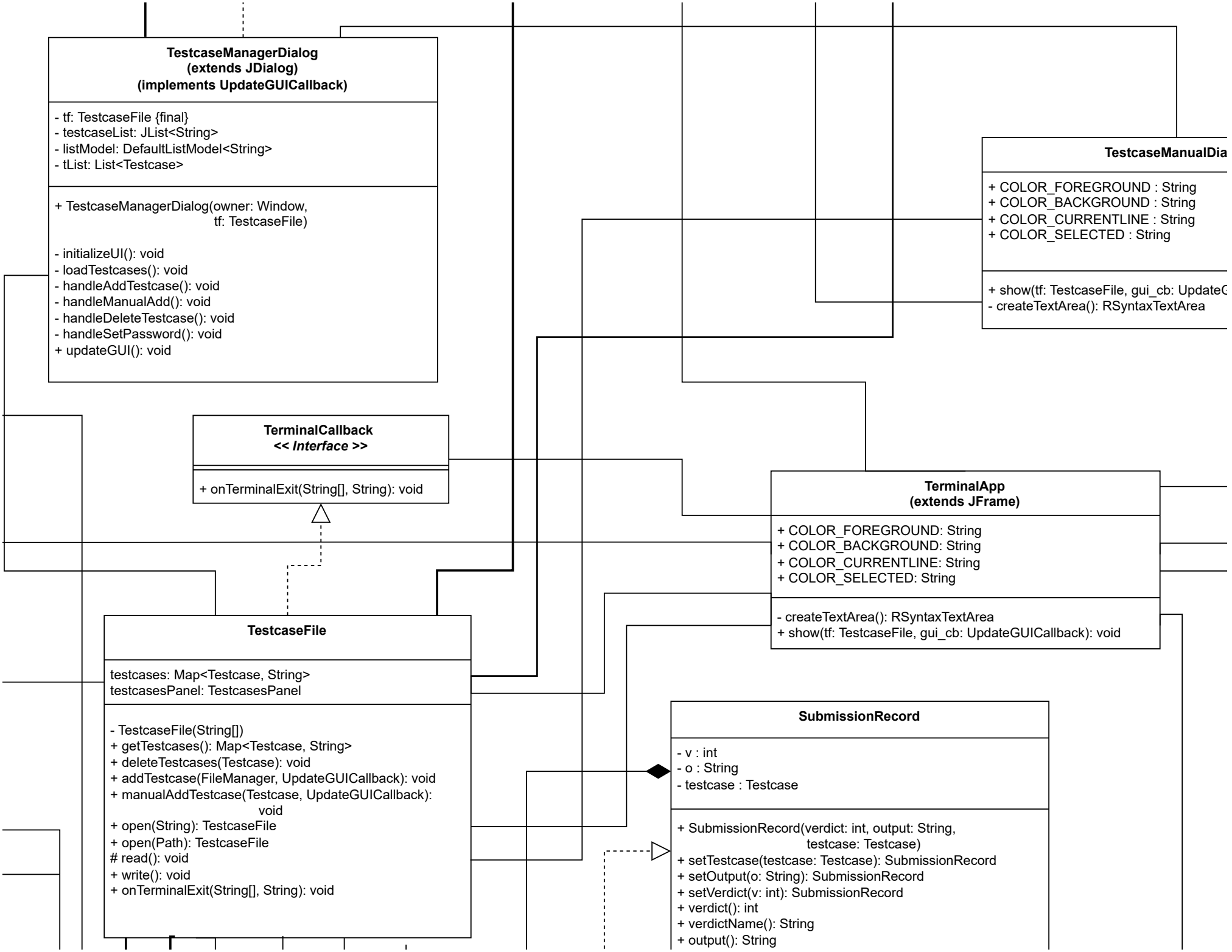

SlaveManager
+ slaveWorkers: SlaveDispatcher
+ romanArmy: ArmyDispatcher
Main
+ main(args:String[]): void
FileExplorer
(extends JPanel)
- fe_tree: JTree {static}
- fileManager: FileManager {static}
- dTextArea: RSyntaxTextArea
- contextMenu: JPopupMenu
- renameItem: JMenuItem
- createFolderItem: JMenuItem
- deleteItem: JMenuItem
- addFileItem: JMenuItem
- textEditor: TextEditor
- selectedFile: SFile
- testcaseFile: TestcaseFile
- fe_instance: FileExplorer {static}
+ FileExplorer(rootDir: String,
editorTextArea: RSyntaxTextArea,
extEditor: TextEditor)
- initializeBackend(rootDir: String): void
- initializeComponents(): void
- setupLayout(): void
- setupEventListeners(): void
- showContextMenu(e: MouseEvent): void
- findNodeByPath(targetPath: Path): DefaultMutableTreeNode {static}
- recursivelyAddNodes(parentNode: DefaultMutableTreeNode,
parentPath: Path): void {static}
- addFileNode(parentNode: DefaultMutableTreeNode,
childPath: Path): void {static}
TextEditor
(extends JPanel)
- runCodeButton: JButton
- addFileButton: JButton
- createButton: JButton
- openFolderButton: JButton
- createFolderButton: JButton
- setEntryPointButton: RoundedButton
- submitCodeButton: JButton
- codeArea: RSyntaxTextArea
- document: StyledDocument
- languageSelectDropdown: JComboBox<String>
- fileExplorerPanel: FileExplorer
- actualOutputArea: JTextPane
- expectedOutputArea: JTextPane
- importTestcaseButton: JButton
- manageTestcaseButton: JButton
- matchStyle: SimpleAttributeSet
- mismatchStyle: SimpleAttributeSet
- excessStyle: SimpleAttributeSet
- defaultStyle: SimpleAttributeSet
- oldLanguage: String (static)
- testcasesPanel: TestcasesPanel
- diffMenu: JPanel
- tcMenu: JPanel
- is_diff_open: boolean
- actual_null_doc: StyledDocument
- expected_null_doc: StyledDocument
- instance: TextEditor (static)
- mainSplit: JSplitPane
- centerRightSplit: JSplitPane
- BUFFER_MAX_CHARS: int

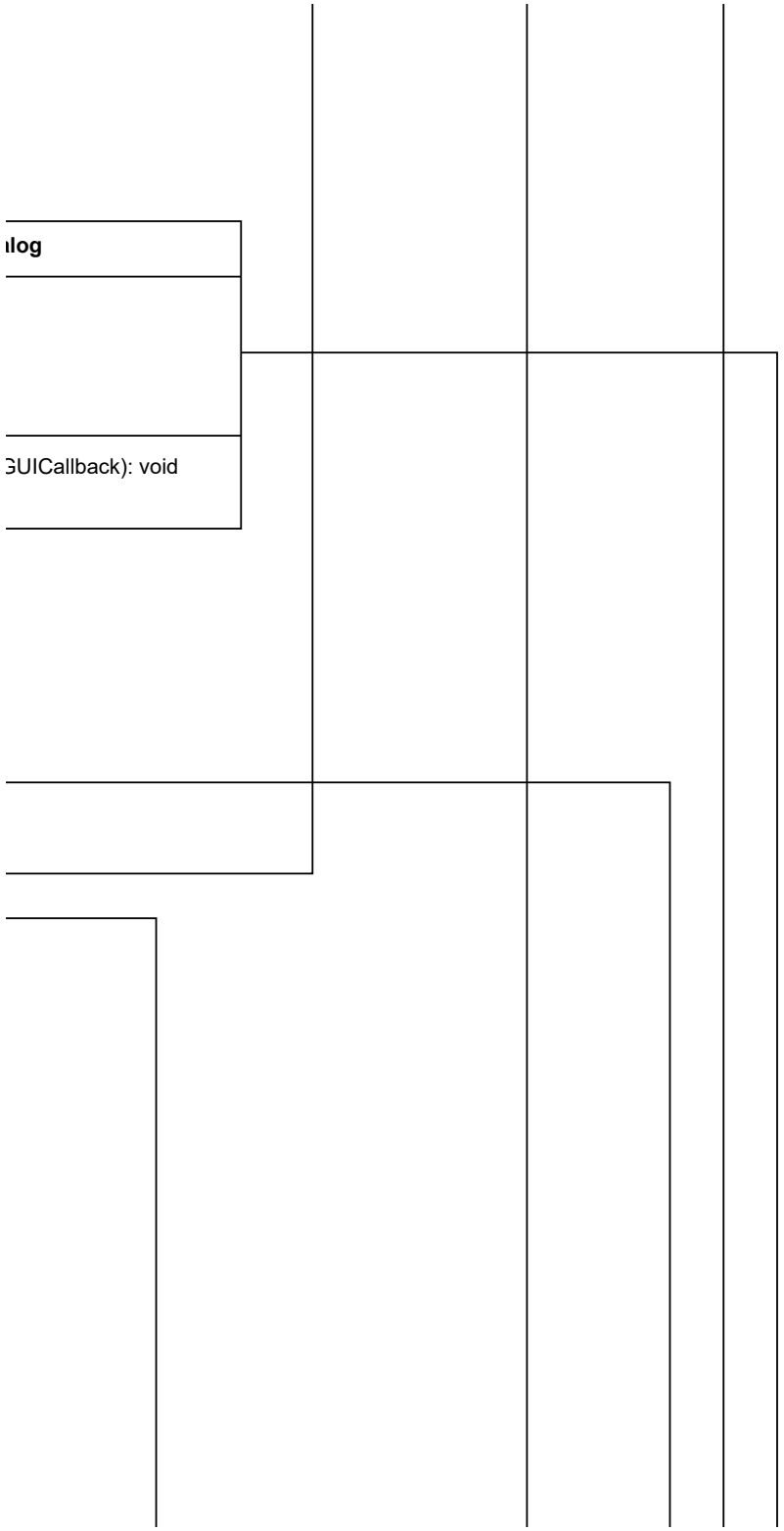

TestcasesPanel
(extends JPanel)
- instance: TestcasesPanel
+ gbc: GridBagConstraints
+ testcaseInsets: Insets
+ monitorWidth: int
+ monitorHeight: int
+ mainPanel: JPanel
+ file: TestcaseFile
+ activeTestcases: Map<testcase, TCEntry>
+ activeBtns: ArrayList<TestcaseButton>
+ entryPool: Deque<TCEntry>
- clearActiveTestcases(): void
+ getActiveTestcases(): Map<Testcase, TCEntry>
+ setTestcaseFile(TestcaseFile): void
- helperAddTestcase(TestEditor, Testcase, int): void
- createDoc(): StyledDocument
- addTestcase(TextEditor, Testcase, TCEntry, int) TestcaseButton
+ addTestcaseCallback(Testcase): void
+ removeTestcaseCallback(Testcase): void
+ getInstance(): TestcasesPanel
- TestcasesPanel()
MainMenu
(extends JPanel)
- title: JLabel
- description: JLabel
- start: JLabel
- openFolder: JButton
+ MainMenu()
TCEntry
+ btn: TestcaseButton
+ actualDoc: StyledDocument
+ expectedDoc: StyledDocument
+ TCEntry(TestcaseButton, StyledDocument, StyledDocument)
UpdateGUICallback
<<Interface>>
+ updateGUI(): void

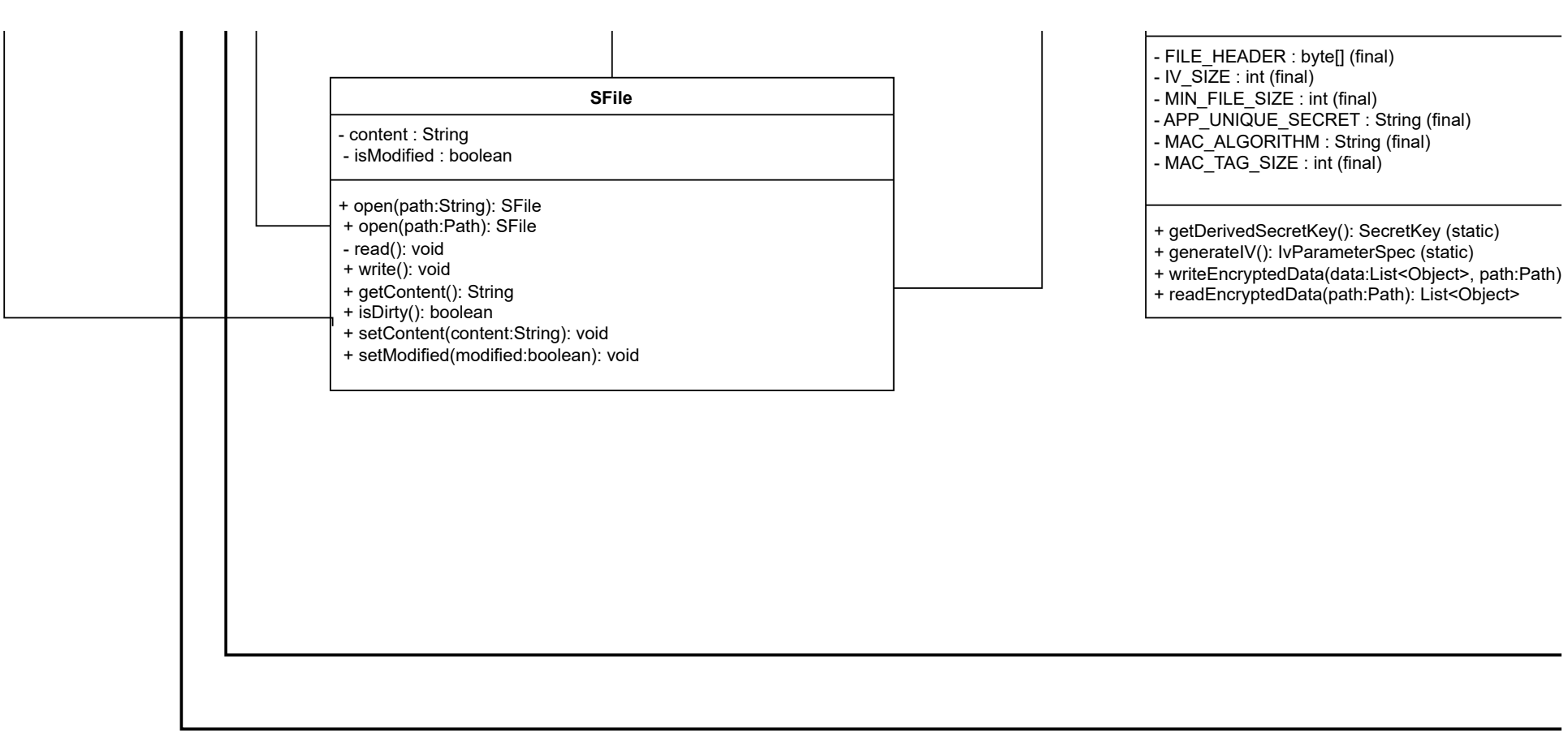


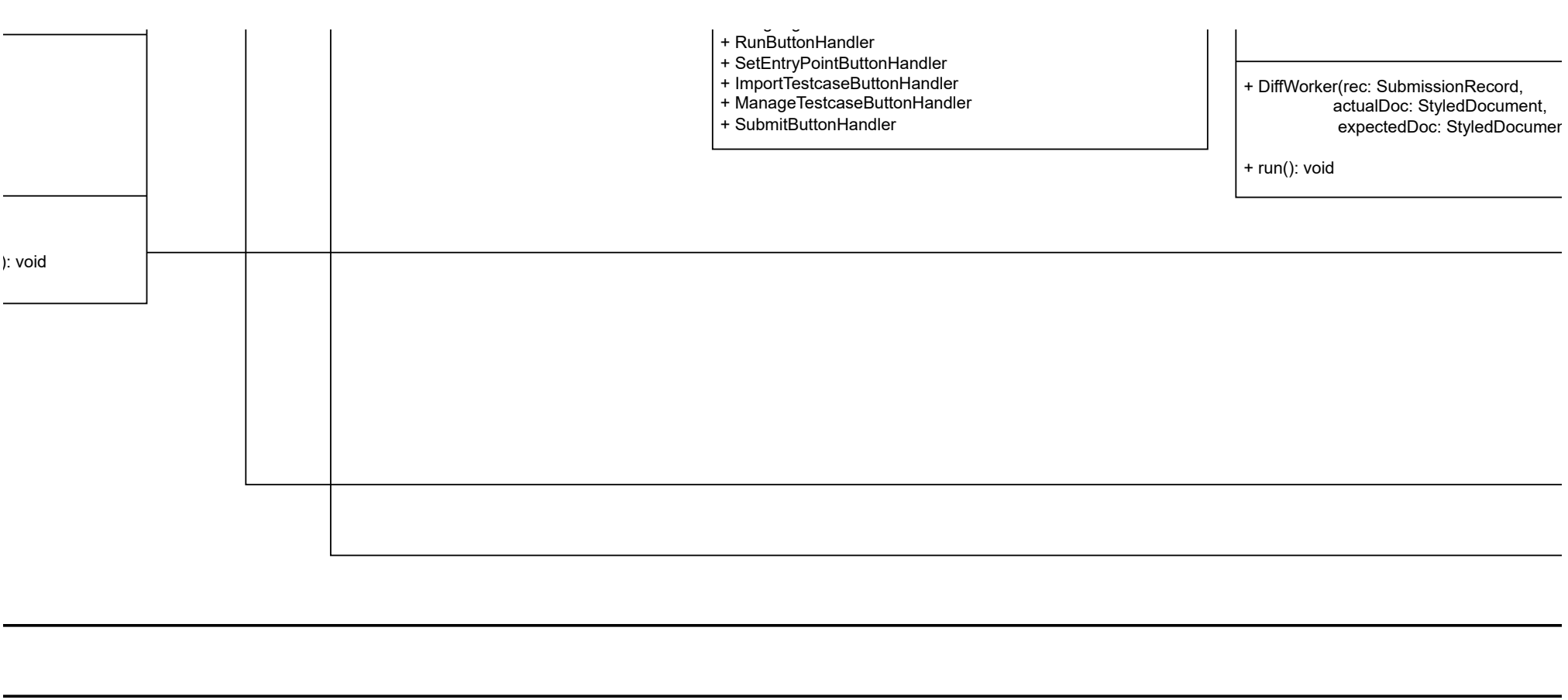

+ setLanguage(language: String): FileManager
+ setRootdir(rootPath: Path): FileManager
+ setRootdir(rootDir: String): FileManager
+ update(): FileManager
+ setAll(rootPath: Path, language: String): FileManager
+ setAll(rootDir: String, language: String): FileManager
+ getRootdir(): Path
+ getLanguage(): String
+ getFiles(): ArrayList<SFile>
+ getLanguageFiles(): ArrayList<SFile>
+ getCurrentFile(): SFile
+ getCurrentFileStringPath(): String
+ setCurrentFile(currentFile: SFile): void
+ getRelativePath(sfile: SFile): Path
+ saveAll(): void
+ openNewFolder(): void
- getAllowedExtensions(lang:String): Set<String>
- listAllContents(rootDir: Path,
allowed_extensions: Set<String>): void
FileWatcher
(implements Runnable)
- fmInstance : FileManager
- feInstance : FileExplorer
- filePaths : List<Path>
- rootDir : Path
- watcher : WatchService
- keys : ConcurrentHashMap<WatchKey, Path>
+ FileWatcher(fm: FileManager,
fe: FileExplorer) throws IOException
+ run() : void
+ closeAndCleanup() : void throws IOException
- registerAll(start:Path) : void throws IOException
- shouldIgnore(dir:Path) : boolean
- handleCreate(path:Path) : void
- handleDelete(deletedPath:Path) : void
CCFile
<< abstract >>
- loadedFiles : Map<Path,CCFile> (static)
path : Path
CCFile(filepath:String)
CCFile(path:Path)
+ getCached(path:String): CCFile (static)
+ getCached(path:Path): CCFile (static)
+ delete(): void
+ getPath(): Path
+ getStringPath(): String
+ setPath(newPath:Path): void
# read(): void (abstract)
+ write(): void (abstract)
CrypticWriter

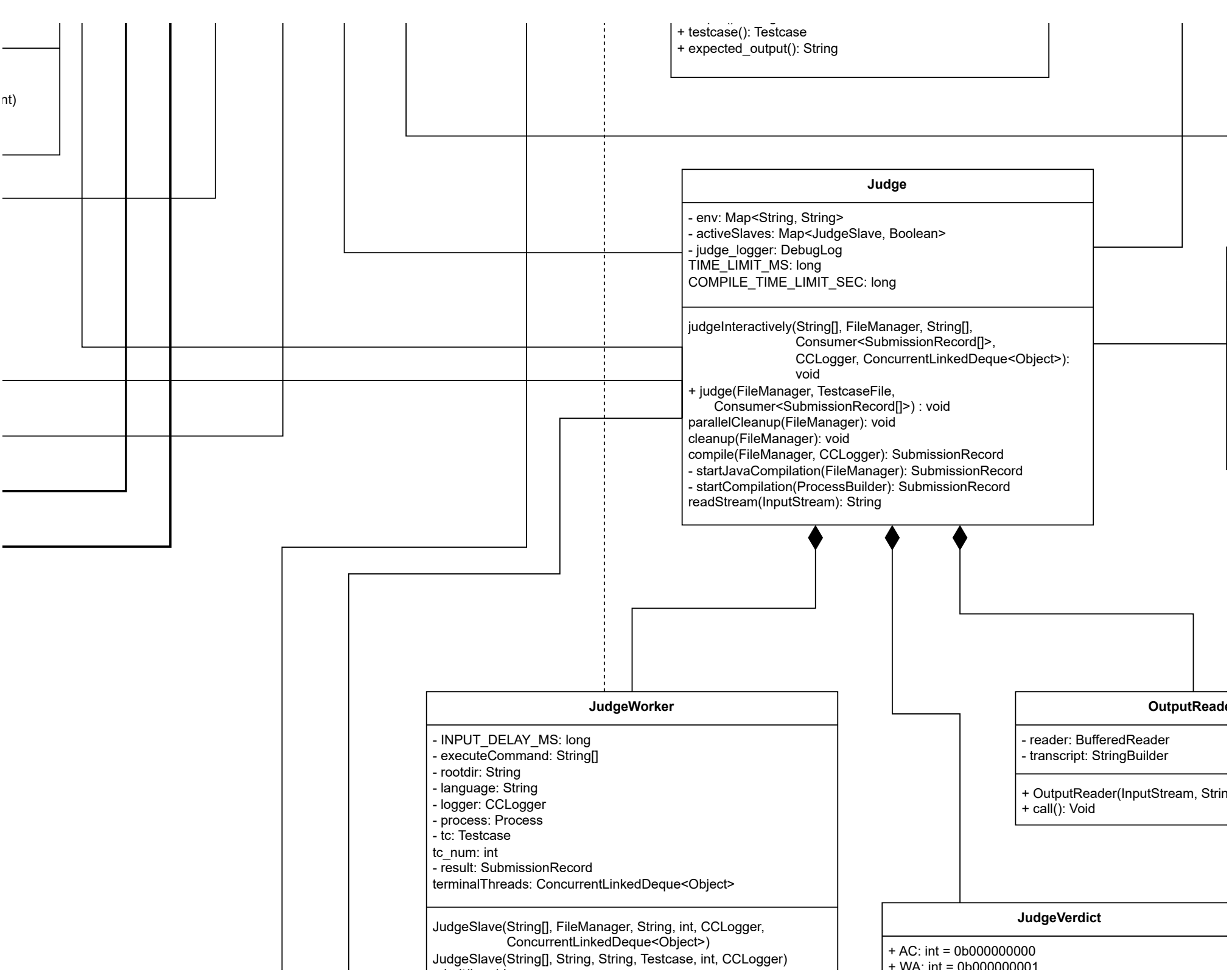

- addFolderNode(parentNode: DefaultMutableTreeNode,
childPath: Path): void {static}
- loadFileContent(sfile: SFile): void
+ updateRootDirectory(newRootDir: String): void
+ handleCreateFolderAction(): void
+ buildFileTree(): void {static}
+ addFileNodeToTree(filePath: Path): void {static}
+ removeFileNodeFromTree(filePath: Path): void {static}
+ resolveNodeToPath(node: DefaultMutableTreeNode): Path
+ getFileManager(): FileManager
+ getSelectedFile(): SFile
+ setSelectedFile(newFile: SFile): void
+ setSyntaxHightlighting(path: String): void
+ getTestcaseFile(): TestcaseFile
+ setTestcaseFile(newFile: TestcaseFile): void
+ getSelectedNode(): DefaultMutableTreeNode
+ getFeTree(): JTree
+ getInstance(): FileExplorer {static}
+ getFileExtension(filePath: String): String {static}
+ getInstance(): TextEditor
- TextEditor()
+ setAll(folderPath: String,
mainMenu: MainMenu): TextEditor
+ setDiffMenuDoc(actual: StyledDocument,
expected: StyledDocument): void
+ openDiffMenu(): void
+ closeDiffMenu(): void
+ saveCurrentFileContent(): void
+ handleAddFileAction(): void
+ getCurrentSelectedLanguage(): String
+ setTextArea(ok: boolean): void
+ makeCoolAndNormal(): void (static)
+ getSetEntryPointButton(): JButton
- initializeComponents(): void
- initializeBackend(): void
- initializeStyles(): void
- setupLayout(): void
- setupEventListeners(): void
- displayActualDiff(actualLines: String[],
expectedLines: String[],
doc: StyledDocument): void
- displayExpectedDiff(actualLines: String[],
expectedLines: String[],
doc: StyledDocument): void
- create_first_panel(): JPanel
- create_second_panel(): JPanel
- create_third_panel_Container(): JPanel
- getScaledIcon(imageUrl: URL, width: int,
height: int): ImageIcon
- configureToolbarButton(button: JButton): void
- canProceedRunCode(): boolean (static)
ComponentHandler
(implements ActionListener)
<< abstract >>
- textEditor: TextEditor
+ ComponentHandler(textEditor: TextEditor)
+ actionPerformed(e: ActionEvent) : void
+ getTextEditor(): TextEditor
Event Handlers
(Static Inner Classes extending ComponentHandler)
+ OpenFolderButtonHandler
+ AddFileButtonHandler
+ LanguageSelectHandler
DiffWorker
(implements Runnable)
- actual: String
- expected: String
- actualDoc: StyledDocument
- expectedDoc: StyledDocument

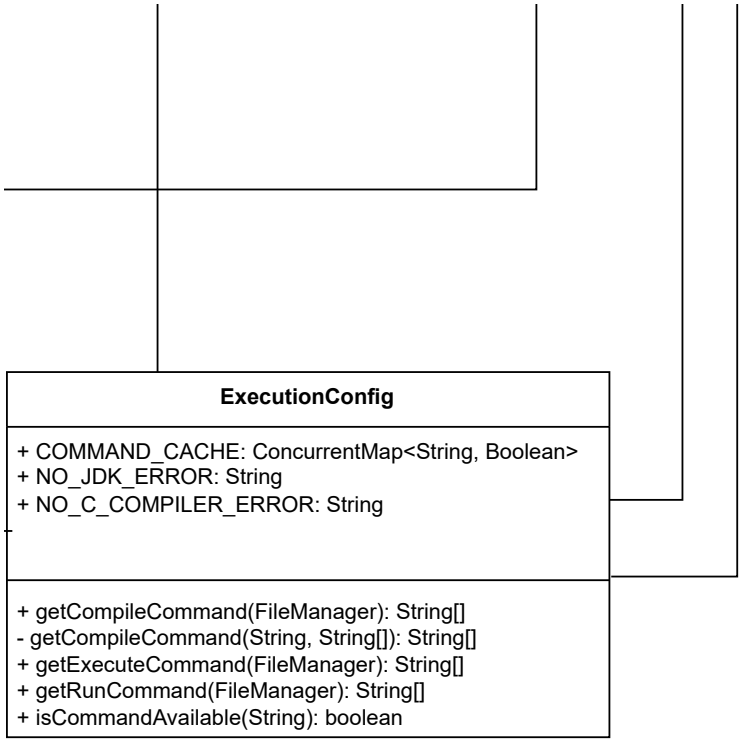

TestcaseManagerDialog
(extends JDialog)
(implements UpdateGUICallback)
- tf: TestcaseFile {final}
- testcaseList: JList<String>
- listModel: DefaultListModel<String>
- tList: List<Testcase>
+ TestcaseManagerDialog(owner: Window,
tf: TestcaseFile)
- initializeUI(): void
- loadTestcases(): void
- handleAddTestcase(): void
- handleManualAdd(): void
- handleDeleteTestcase(): void
- handleSetPassword(): void
+ updateGUI(): void
TestcaseManualDia
+ COLOR_FOREGROUND : String
+ COLOR_BACKGROUND : String
+ COLOR_CURRENTLINE : String
+ COLOR_SELECTED : String
+ show(tf: TestcaseFile, gui_cb: UpdateG
- createTextArea(): RSyntaxTextArea
TerminalCallback
<< Interface >>
+ onTerminalExit(String[], String): void
TerminalApp
(extends JFrame)
+ COLOR_FOREGROUND: String
+ COLOR_BACKGROUND: String
+ COLOR_CURRENTLINE: String
+ COLOR_SELECTED: String
- createTextArea(): RSyntaxTextArea
+ show(tf: TestcaseFile, gui_cb: UpdateGUICallback): void
TestcaseFile
testcases: Map<Testcase, String>
testcasesPanel: TestcasesPanel
- TestcaseFile(String[])
+ getTestcases(): Map<Testcase, String>
+ deleteTestcases(Testcase): void
+ addTestcase(FileManager, UpdateGUICallback): void
+ manualAddTestcase(Testcase, UpdateGUICallback):
void
+ open(String): TestcaseFile
+ open(Path): TestcaseFile
# read(): void
+ write(): void
+ onTerminalExit(String[], String): void
SubmissionRecord
- v : int
- o : String
- testcase : Testcase
+ SubmissionRecord(verdict: int, output: String,
testcase: Testcase)
+ setTestcase(testcase: Testcase): SubmissionRecord
+ setOutput(o: String): SubmissionRecord
+ setVerdict(v: int): SubmissionRecord
+ verdict(): int
+ verdictName(): String
+ output(): String

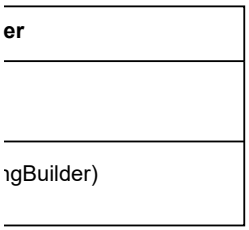

log
GUICallback): void

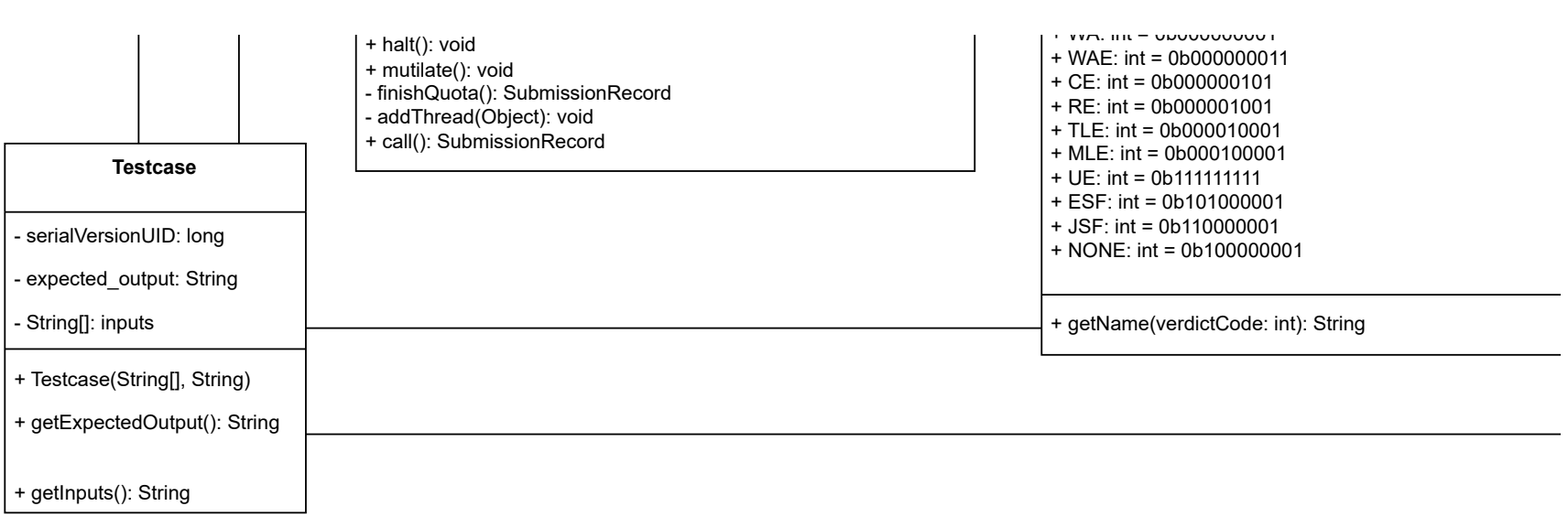

SFile
- content : String
- isModified : boolean
+ open(path:String): SFile
+ open(path:Path): SFile
- read(): void
+ write(): void
+ getContent(): String
+ isDirty(): boolean
+ setContent(content:String): void
+ setModified(modified:boolean): void
- FILE_HEADER : byte[] (final)
- IV_SIZE : int (final)
- MIN_FILE_SIZE : int (final)
- APP_UNIQUE_SECRET : String (final)
- MAC_ALGORITHM : String (final)
- MAC_TAG_SIZE : int (final)
+ getDerivedSecretKey(): SecretKey (static)
+ generateIV(): IvParameterSpec (static)
+ writeEncryptedData(data:List<Object>, path:Path)
+ readEncryptedData(path:Path): List<Object>

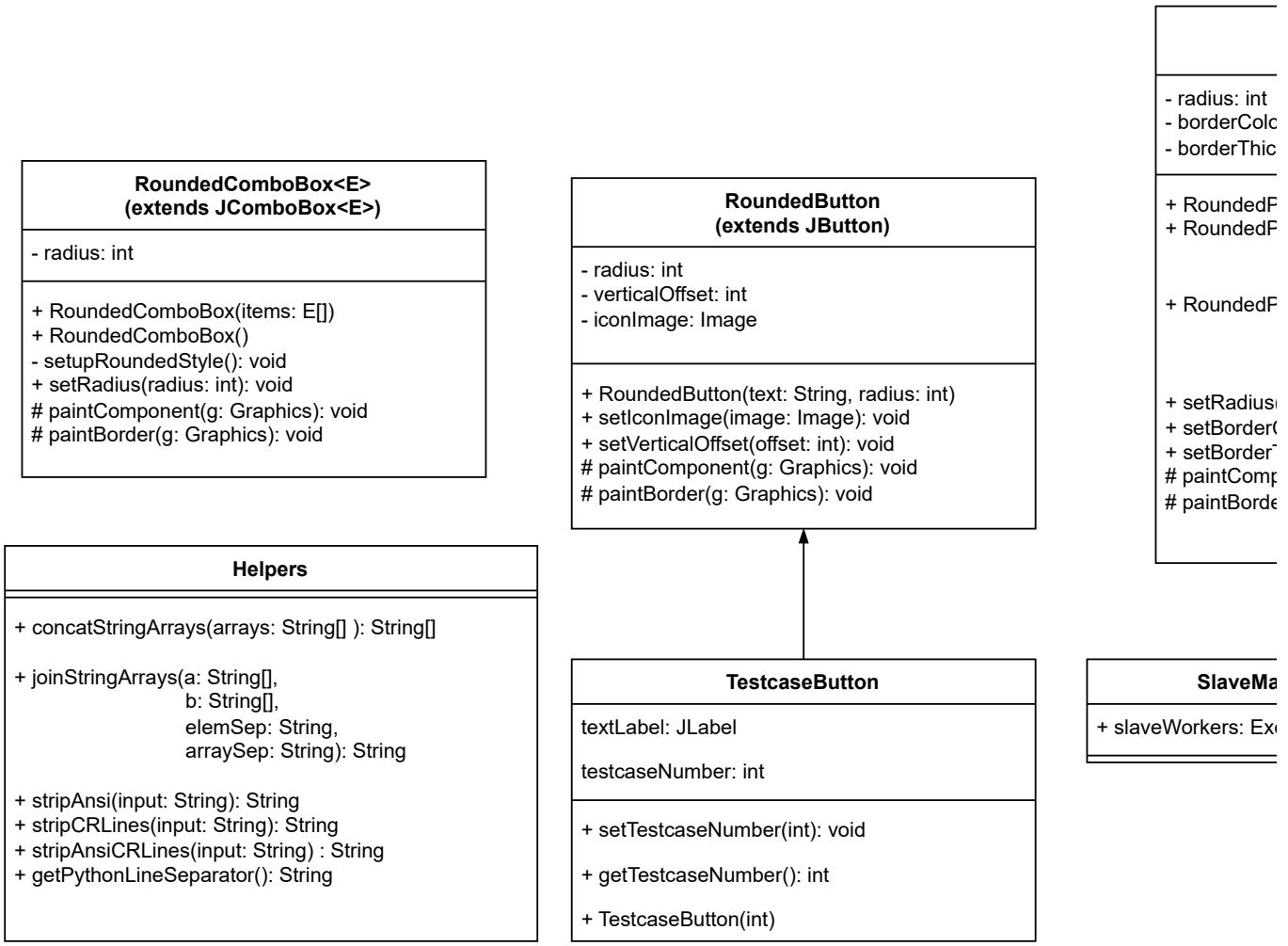

+ RunButtonHandler
+ SetEntryPointButtonHandler
+ ImportTestcaseButtonHandler
+ ManageTestcaseButtonHandler
+ SubmitButtonHandler
+ DiffWorker(rec: SubmissionRecord,
actualDoc: StyledDocument,
expectedDoc: StyledDocumer
+ run(): void
): void

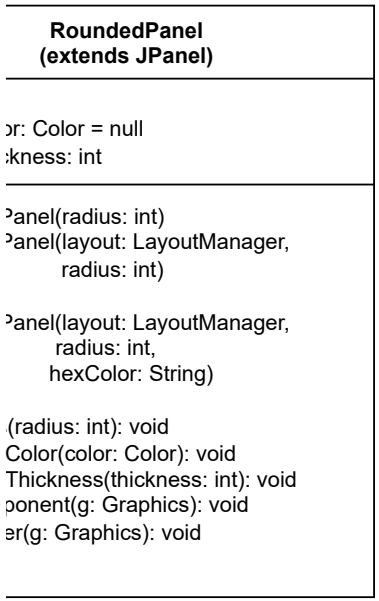

+ testcase(): Testcase
+ expected_output(): String
Judge
- env: Map<String, String>
- activeSlaves: Map<JudgeSlave, Boolean>
- judge_logger: DebugLog
TIME_LIMIT_MS: long
COMPILE_TIME_LIMIT_SEC: long
judgeInteractively(String[], FileManager, String[],
Consumer<SubmissionRecord[]>,
CCLogger, ConcurrentLinkedDeque<Object>):
void
+ judge(FileManager, TestcaseFile,
Consumer<SubmissionRecord[]>) : void
parallelCleanup(FileManager): void
cleanup(FileManager): void
compile(FileManager, CCLogger): SubmissionRecord
- startJavaCompilation(FileManager): SubmissionRecord
- startCompilation(ProcessBuilder): SubmissionRecord
readStream(InputStream): String
JudgeWorker
- INPUT_DELAY_MS: long
- executeCommand: String[]
- rootdir: String
- language: String
- logger: CCLogger
- process: Process
- tc: Testcase
tc_num: int
- result: SubmissionRecord
terminalThreads: ConcurrentLinkedDeque<Object>
JudgeSlave(String[], FileManager, String, int, CCLogger,
ConcurrentLinkedDeque<Object>)
JudgeSlave(String[], String, String, Testcase, int, CCLogger)
OutputReade
- reader: BufferedReader
- transcript: StringBuilder
+ OutputReader(InputStream, Strin
+ call(): Void
JudgeVerdict
+ AC: int = 0b000000000
+ WA: int = 0b000000001

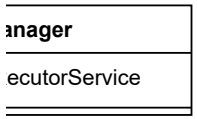

ExecutionConfig
+ COMMAND_CACHE: ConcurrentMap<String, Boolean>
+ NO_JDK_ERROR: String
+ NO_C_COMPILER_ERROR: String
+ getCompileCommand(FileManager): String[]
- getCompileCommand(String, String[]): String[]
+ getExecuteCommand(FileManager): String[]
+ getRunCommand(FileManager): String[]
+ isCommandAvailable(String): boolean


er
ıgBuilder)


+ halt(): void
+ mutilate(): void
- finishQuota(): SubmissionRecord
- addThread(Object): void
+ call(): SubmissionRecord
+ WAE: int = 0b000000011
+ CE: int = 0b000000101
+ RE: int = 0b000001001
+ TLE: int = 0b000010001
+ MLE: int = 0b000100001
+ UE: int = 0b111111111
+ ESF: int = 0b101000001
+ JSF: int = 0b110000001
+ NONE: int = 0b100000001
+ getName(verdictCode: int): String
Testcase
- serialVersionUID: long
- expected_output: String
- String[]: inputs
+ Testcase(String[], String)
+ getExpectedOutput(): String
+ getInputs(): String


RoundedComboBox<E>
(extends JComboBox<E>)
- radius: int
+ RoundedComboBox(items: E[])
+ RoundedComboBox()
- setupRoundedStyle(): void
+ setRadius(radius: int): void
# paintComponent(g: Graphics): void
# paintBorder(g: Graphics): void
RoundedButton
(extends JButton)
- radius: int
- verticalOffset: int
- iconImage: Image
+ RoundedButton(text: String, radius: int)
+ setIconImage(image: Image): void
+ setVerticalOffset(offset: int): void
# paintComponent(g: Graphics): void
# paintBorder(g: Graphics): void
- radius: int
- borderColo
- borderThic
+ RoundedP
+ RoundedP
+ RoundedP
+ setRadius
+ setBorder
+ setBorder
# paintComp
# paintBorde
Helpers
+ concatStringArrays(arrays: String[]): String[]
+ joinStringArrays(a: String[],
b: String[],
elemSep: String,
arraySep: String): String
+ stripAnsi(input: String): String
+ stripCRLines(input: String): String
+ stripAnsiCRLines(input: String) : String
+ getPythonLineSeparator(): String
TestcaseButton
textLabel: JLabel
testcaseNumber: int
+ setTestcaseNumber(int): void
+ getTestcaseNumber(): int
+ TestcaseButton(int)
SlaveMa
+ slaveWorkers: Ex


RoundedPanel
(extends JPanel)
or: Color = null
kness: int
Panel(radius: int)
Panel(layout: LayoutManager,
radius: int)
Panel(layout: LayoutManager,
radius: int,
hexColor: String)
(radius: int): void
Color(color: Color): void
Thickness(thickness: int): void
ponent(g: Graphics): void
er(g: Graphics): void


anager
ecutorService