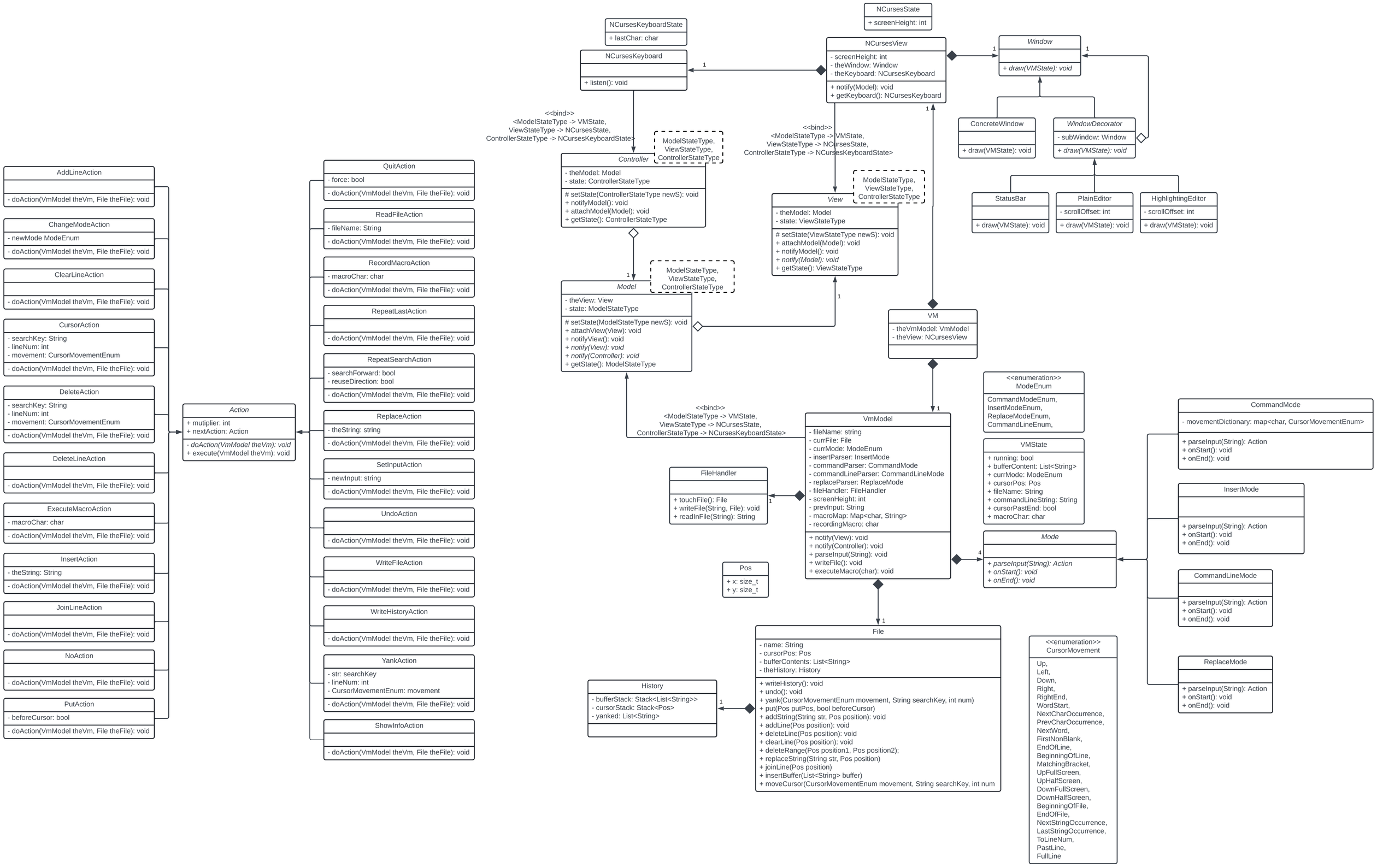
NCursesKeyboardState
+ lastChar: char
NCursesKeyboard
+ listen(): void
NCursesState
+ screenHeight: int
NCursesView
- screenHeight: int
- theWindow: Window
- theKeyboard: NCursesKeyboard
+ notify(Model): void
+ getKeyboard(): NCursesKeyboard
Window
+ draw(VMState): void
ConcreteWindow
+ draw(VMState): void
WindowDecorator
- subWindow: Window
+ draw(VMState): void
StatusBar
+ draw(VMState): void
PlainEditor
- scrollOffset: int
+ draw(VMState): void
HighlightingEditor
- scrollOffset: int
+ draw(VMState): void
<<bind>>
<ModelStateType -> VMState,
ViewStateType -> NCursesState,
ControllerStateType -> NCursesKeyboardState>
ModelStateType,
ViewStateType,
ControllerStateType
Controller
- theModel: Model
- state: ControllerStateType
setState(ControllerStateType newS): void
+ notifyModel(): void
+ attachModel(Model): void
+ getState(): ControllerStateType
View
- theModel: Model
- state: ViewStateType
setState(ViewStateType newS): void
+ attachModel(Model): void
+ notifyModel(): void
+ notify(Model): void
+ getState(): ViewStateType
Model
- theView: View
- state: ModelStateType
setState(ModelStateType newS): void
+ attachView(View): void
+ notifyView(): void
+ notify(View): void
+ notify(Controller): void
+ getState(): ModelStateType
VM
- theVmModel: VmModel
- theView: NCursesView
<<enumeration>>
ModeEnum
CommandModeEnum,
InsertModeEnum,
ReplaceModeEnum,
CommandLineEnum,
VmModel
- fileName: string
- currFile: File
- currMode: ModeEnum
- insertParser: InsertMode
- commandParser: CommandMode
- commandLineParser: CommandLineMode
- replaceParser: ReplaceMode
- fileHandler: FileHandler
- screenHeight: int
- prevInput: String
- macroMap: Map<char, String>
- recordingMacro: char
+ notify(View): void
+ notify(Controller): void
+ parseInput(String): void
+ writeFile(): void
+ executeMacro(char): void
VMState
+ running: bool
+ bufferContent: List<String>
+ currMode: ModeEnum
+ cursorPos: Pos
+ fileName: String
+ commandLineString: String
+ cursorPastEnd: bool
+ macroChar: char
FileHandler
+ touchFile(): File
+ writeFile(String, File): void
+ readInFile(String): String
Pos
+ x: size_t
+ y: size_t
Mode
+ parseInput(String): Action
+ onStart(): void
+ onEnd(): void
CommandMode
- movementDictionary: map<char, CursorMovementEnum>
+ parseInput(String): Action
+ onStart(): void
+ onEnd(): void
InsertMode
+ parseInput(String): Action
+ onStart(): void
+ onEnd(): void
CommandLineMode
+ parseInput(String): Action
+ onStart(): void
+ onEnd(): void
ReplaceMode
+ parseInput(String): Action
+ onStart(): void
+ onEnd(): void
File
- name: String
- cursorPos: Pos
- bufferContents: List<String>
- theHistory: History
+ writeHistory(): void
+ undo(): void
+ yank(CursorMovementEnum movement, String searchKey, int num)
+ put(Pos putPos, bool beforeCursor)
+ addString(String str, Pos position): void
+ addLine(Pos position): void
+ deleteLine(Pos position): void
+ clearLine(Pos position): void
+ deleteRange(Pos position1, Pos position2);
+ replaceString(String str, Pos position)
+ joinLine(Pos position)
+ insertBuffer(List<String> buffer)
+ moveCursor(CursorMovementEnum movement, String searchKey, int num
History
- bufferStack: Stack<List<String>>
- cursorStack: Stack<Pos>
- yanked: List<String>
<<enumeration>>
CursorMovement
Up,
Left,
Down,
Right,
RightEnd,
WordStart,
NextCharOccurrence,
PrevCharOccurrence,
NextWord,
FirstNonBlank,
EndOfLine,
BeginningOfLine,
MatchingBracket,
UpFullScreen,
UpHalfScreen,
DownFullScreen,
DownHalfScreen,
BeginningOfFile,
EndOfFile,
NextStringOccurrence,
LastStringOccurrence,
ToLineNum,
PastLine,
FullLine
Action
+ mutiplier: int
+ nextAction: Action
- doAction(VmModel theVm): void
+ execute(VmModel theVm): void
AddLineAction
- doAction(VmModel theVm, File theFile): void
ChangeModeAction
- newMode ModeEnum
- doAction(VmModel theVm, File theFile): void
ClearLineAction
- doAction(VmModel theVm, File theFile): void
CursorAction
- searchKey: String
- lineNum: int
- movement: CursorMovementEnum
- doAction(VmModel theVm, File theFile): void
DeleteAction
- searchKey: String
- lineNum: int
- movement: CursorMovementEnum
- doAction(VmModel theVm, File theFile): void
DeleteLineAction
- doAction(VmModel theVm, File theFile): void
ExecuteMacroAction
- macroChar: char
- doAction(VmModel theVm, File theFile): void
InsertAction
- theString: String
- doAction(VmModel theVm, File theFile): void
JoinLineAction
- doAction(VmModel theVm, File theFile): void
NoAction
- doAction(VmModel theVm, File theFile): void
PutAction
- beforeCursor: bool
- doAction(VmModel theVm, File theFile): void
QuitAction
- force: bool
- doAction(VmModel theVm, File theFile): void
ReadFileAction
- fileName: String
- doAction(VmModel theVm, File theFile): void
RecordMacroAction
- macroChar: char
- doAction(VmModel theVm, File theFile): void
RepeatLastAction
- doAction(VmModel theVm, File theFile): void
RepeatSearchAction
- searchForward: bool
- reuseDirection: bool
- doAction(VmModel theVm, File theFile): void
ReplaceAction
- theString: string
- doAction(VmModel theVm, File theFile): void
SetInputAction
- newInput: string
- doAction(VmModel theVm, File theFile): void
UndoAction
- doAction(VmModel theVm, File theFile): void
WriteFileAction
- doAction(VmModel theVm, File theFile): void
WriteHistoryAction
- doAction(VmModel theVm, File theFile): void
YankAction
- str: searchKey
- lineNum: int
- CursorMovementEnum: movement
- doAction(VmModel theVm, File theFile): void
ShowInfoAction
- doAction(VmModel theVm, File theFile): void