using mapKeyValues = std::map<std::string, std::map<std::string, std::string>>; UI **INIEditor** + INIEditor() + UI(INIEditor *); + ~INIEditor(): Has custom implementation for unexpected exits editor: INIEditor * + vector<string> getLines() const; void clearCin() const; + vector<string> getNumberOfLines() const; bool confirmInputPrompt() const; + void listSections() const; string inputName() const; + string getCurrentFilename() const; unsigned int inputNum() const; + pair<string, string> getKeyValuePair(unsigned int) const; vector<string> detectINIs() const; + const mapKeyValues * getPtrToKeyValuesMap() const; void mainMenu() const; + void clearAll(); void createNew() const; + void deleteLine(unsigned int); void loadFile() const; + void parseMapFromLines(); void workWithFile() const; + void parseWorkingFile(); void lineMenu(unsigned int) const; + void replaceEntireLine(int, string); void deleteLine(unsigned int) const; + void replaceLineByKeyValuePair(int, pair<string, string>); void editLine(unsigned int) const; + void setExpectedExit(); + void setWorkingFile(string); + void writeLinesToFile(); - lines: vector<string> - sectionsKeyValues: map<string, map<string, string>> - workingFile: string - isUnexectedExit: bool void replaceLine(int, string&); void saveCurrentLines(string); vector<string> parseFileLinesToStringVector(string) const; pair<string, string> getKeyValuePair(string) const; - mapkeyValues * getSectionsKeyValuePairsMap(vector<string>) const;