pt.tecnico.uilib.form

<> Field

- -prompt: String
- -clear: boolean
- # -value: Type

- + clear(): void
- # dirty(): void
- + cleared(): boolean
- + isReadOnly: boolean
- <> Parse(in: String): boolean

FieldBoolean

- + Parse(in: String): boolean

FieldReal

- + Parse(in: String): boolean

FieldInteger

- + Parse(in: String): boolean

FieldString

- + Parse(in: String): boolean

FieldName

- + Parse(in: String): boolean
- + isReadOnly(): boolean

Form

- -ui: Dialog
- -title: String
- -fields: Map<String, Field<?>>

- + entries(): Collection<Field<?>>
- # field(key: String): Field<?>
- + parse(): Form
- + parse(clear: Boolean): Form
- + clear(): void

FormException

- serialVersionUID: long

<<Interface>> Menage

- keyAlreadyExist(form: String, key: String): String
- keyNotFound(form: String, key: String): String
- typeMismatch(form: String, key: String, actualType: String, requestType: String): String

---

pt.tecnico.uilib

Dialog

- <<final>> ACTION_CHANNEL: String
- <<final>> CHANNEL_SWING: String
- <<final>> CHANNEL_NEW_SWING: String
- <<final>> CHANNEL_TEXT
- + UI: Dialog
- -backend: InteractionDriver

- + open(menu: Menu): void
- + fill(form: Form): void
- + render(title: String, text: String): void
- + close(): void

Display

- -ui: Dialog
- -title: String
- -text: StringBuilder

- + add(toAdd: Object): Display
- + addAll(items: Collection<?>): Display
- + addLine(toAdd: Object): Display
- + addNewLine(toAdd: Object, pre: boolean): Display
- + display(): void
- + popup(toPop: Object): void
- + popup(toPop: Collection<?>): void
- + clear(): void

<<Interface>> InteractionDriver

- open(menu: Menu): void
- fill(form: Form): void
- render(title: String, text: String): void
- close(): void

CompositePrintStream

- -streams: Collection<PrintStream>
- -error: boolean

<<final>> add (ps: PrintStream): void
+ checkError(): boolean
+ close(): void
+ flush(): void
+ print(b: boolean): void
+ print(c: char): void
+ print(c: char[]): void
+ print(d: double): void
+ print(f: float): void
+ print(i: int): void
+ print(l: long): void
+ print(ob): Object): void
+ print(str: string): void
+ println(b: boolean): void
+ println(c: char): void
+ println(c: char[]): void
+ println(d: double): void
+ println(f: float): void
+ println(i: int): void
+ println(l: long): void
+ println(ob): Object): void
+ println(str: string): void
+ write(buf: byte[], off: int, len: int): void
+ write(b: int): void
+ write(b: byte[]): void

RuntimeEOFException

SERIALVERSIONUID: long

TEXtInteraction

- -in: BufferedReader
- -out: PrintStream
- -log: PrintStream
- -writeInput: boolean

+ close(): void
+ open(menu: menu): void
+ fill(form: form): void
+ render(title: string, text: string): void
- <<final>> readString(prompt: string): string
- <<final>> readInteger(prompt: string): string

<<Interface>> Message

invalidOperation(err: string): String
invalidOption(): String
ErrorEOF(e: EOFException): String
error IO(e: IOException): String
ErrorInvalidNumber(e: NumberFormatException): String
ErrorEOF(e: RuntimeEOFException): String

inputError(e: FileNotFoundException): String
errorDuringInput(e: IOException): String
EndOfInput(): String
outputError(e: FileNotFoundException): String
logError(e: FileNotFoundException): String

<<Interface>> Prompt

option(): String
exit(): String

<<Interface>> Properties

WRITE_INPUT: String
BOTH_CHANNEL: String
OUTPUT_CHANNEL: String
LOG_CHANNEL: String

pt.tecnico.uilib.menus
Receiver: class
<<abstract>> Command
- _last: boolean
- _title: String
<<final>> _receiver: Receiver
_valid: Predicate<Receiver>
- _FORM: Form
_DISPLAY: Display
+ <<final>> performCommand(): void
# <<abstract>> execute(): void
<<abstract>> CommandException
- SERIALVERSION: long
DoOpenMenu
<<final>> execute(): void
Menu
- _ui: Dialog
- _title: String
- _commands: Command<?>[]
+ size(): int
+ entry(n: int): Command<?>
+ entries(): Command<?>[]
+ open(): void
<<Interface>>
Message
operationFailed(error: String): String
No caso a visibilidade dos atributos ou métodos não for especificada, estes são visíveis unicamente por classes na mesma package
pt.tecnico.uilib.swing
NewSwingInteraction
- SERIALVERSION: long
- MAIN_TITLE: String
- _currentDialogMenu: SwingDialogMenu
+ open(menu: Menu): void
+ fill(form: Form): void
+ render(title: String, text: String): void
+ close(): void
- message(title: String, msg: String): void
<<Interface>>
Prompt
exit(): String
SwingForm
- SERIALVERSIONUID: long
- _inputs: Map<Field<?>, JTextField>
parse(): boolean
SwingInteraction
- SERIALVERSIONUID: long
- MAIN_TITLE: String
+ open(menu: Menu): void
+ fill(form: Form): void
+ render(title: String, text: String): void
+ close(): void
- message(title: String, msg: String): void
SwingDialogForm
- RED_BORDER: Border
- NO_BORDER: Border
- SERIALVERSIONUID: long
- _inputs: Map<Field<?>, JTextField>
- createFormPanel(form: Form): JPanel
+ actionPerformed(event: ActionEvent): void
parse(): boolean
SwingDialogMenu
- SERIALVERSIONUID: long
- _option: int
- _menu: Menu
- _out: JTextArea
- _outputTitleLabel: JLabel
- _buttonMenu: JButton[]
- _selectedButton: JButton
- createOutputPane(): JPanel
+ updateOutput(title: String, output: String): void
- createMenuPanel(): JPanel
+ actionPerformed(event: ActionEvent): void
- updateValidButtons(): void
- executeOption(option: int): void
option(): int
- cleanAndHide(): void
- showErrorMessage(title: String, msg: String): void
SwingMenu
- SERIALVERSIONUID: long
- _option: int
+ actionPerformed(event: ActionEvent): void
option(): int
SwingMessage
- SERIALVERSIONUID: long
<<abstract>> SwingPanel
- SERIALVERSIONUID: long
_end: boolean
_LOCK: Object
+ actionPerformed(event: ActionEvent): void
sleep(millis: int): void
await(): void

Scanned with CamScanner

Declaro por minha honra que ste diagrama foi realizado apenas pelos elementos que constituem o grupo de projeto.

Allan Farmamols

Declaro por minha honra que ste diagrama foi realizado apenas pelos elementos que constituem o grupo de projeto.

Allan Farmamols

Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo de projeto. Lucas Fortunato dos Alves