



`<<abstract>>`
Field

`--prompt: String`
`--clean: Boolean`
`# - value: Type`
`+ prompt(): String`
`+ set(value: Type): void`
`+ value(): Type`
`+ clean(): void`
`# dirty(): void`
`+ cleared(): bool`
`+ is Read Only(): bool`
`+ <<abstract>> parse(im: String): bool`

Form

`+ title: String`
`- fields: Map<String, Field<?>> = new LinkedHashMap<>()`
`+ title(): String`
`+ entries(): Collection<Field<?>>`
`+ field(key: String): Field<?>`
`- add(key: String, im: Field<?>): void`
`+ addBooleanField(key: String, label: String): void`
`+ addStringField(key: String, label: String): void`
`+ addRealField(key: String, label: String): void`
`+ addIntegerField(key: String, label: String): void`
`- get(key: String, type: String): Object`
`+ booleanField(key: String): bool`
`+ stringField(key: String): String`

+ realField (key: String): Double
+ integerField (key: String): Integer
+ parse (): Form
+ parse (clean: bool): Form
+ clean (): void
+ confirm (prompt: String): bool
+ requestInteger (prompt: String): Integer
+ requestReal (prompt: String): Double
+ requestString (prompt: String): String

<< Interface >>

Messages

KeyAlreadyExists (form: String, key: String): String

keyNotFound (form: String, key: String): String

typeMismatch (form: String, key: String, actualType: String, requiredType: String): String

requiredType = String

<<abstract>>

Command

- - lost : bool

- - title : String

- RECEIVER : Receiver

- valid : Predicate<Receiver> = receiver->true

<class>

+ title() : String

+ isLost() : bool

+ isVoid() : bool

+ add Boolean Field (key : String, prompt : String) : void

+ add Real Field (key : String, prompt : String) : void

+ add Integer Field (key : String, prompt : String) : void

+ add String Field (key : String, prompt : String) : void

+ boolean Field (key : String) : bool

+ real Field (key : String) : Double

+ integer Field (key : String) : Integer

+ string Field (key : String) : String

+ << final >> perform Command () : void

<< abstract >> execute () : void

Do Open Menu

<< final >> execute () : void

<< interface >>
Message

OperationFailed (error : String)

Menu

- - title : String

+ title () : String

+ size () : int

+ entry (n : int) : Command<?>

+ entries () : Command<?>[]

+ open () : void

Composite Print Stream

- - streams: Collection<PrintStream> = new ArrayList()

- - error: bool = false

<< final >> add (ps: PrintStream) : void

+ checkError () : bool

+ close () : void

+ flush () : void

+ print (b: bool) : void

+ print (c: char) : void

+ print (c: char []) : void

+ print (d: double) : void

+ print (i: int) : void

+ print (l: long) : void

+ print (obj: Object) : void

+ print (str: String) : void

+ println (b: bool) : 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 (obj: Object) : void

+ println (str: String) : void

setError () : void

+ write (buf: byte [], off: int, len: int) : void

+ write (b: int) : void
+ write (b: byte[]) : void

<< interface >>

Message

invalid Operation (error: String) : String
invalid Option () : String
error EOF (e: EOFException) : String
error IO (ioe: IOException) : String
error Invalid Number (e: NumberFormatException) : String
error REOF (e: RuntimeEOFException e) : String
input Error (e: FileNotFoundException) : String
error Closing Input (e: IOException) : String
end Of Input () : String
output Error (e: FileNotFoundException) : String
log Error (e: FileNotFoundException) : String

<< interface >>

Prompt

option () : String
exit () : String

<< interface >>

Properties

WRITE_INPUT: String = "writeInput"
BOTH_CHANNEL: String = "both"
INPUT_CHANNEL: String = "in"
OUTPUT_CHANNEL: String = "out"
LOG_CHANNEL: String = "log"

Brutaline EDException

Text Interaction

- - in : BufferedReader = new BufferedReader(new InputStreamReader(S))
- - out : PrintStream = System.out
- - log : PrintStream = null
- - writeInput : bool

- + open(menu : Menu) : void
- + fill (form: Form) : void
- + render (title: String , text: String) : void
- << final >> readString (prompt : String) : String
- << final >> readInteger (prompt: String) : int

Melody

<< interface >>

Interaction Driven

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

Dialog

```
- <<final>> ACTION CHANNEL = "ui"           String  
- <<final>> CHANNEL_SWING = "swing": String = "swing"  
- <<final>> CHANNEL_NEW_SWING: String = "new swing"  
- <<final>> CHANNEL_TEXT: String = "text"  
+ UI: Dialog = new Dialog()
```

UI Methods

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

Display

```
- - title: String = ""  
- - text: StringBuilder = new StringBuilder()  
+ add (toAdd: Object): Display  
+ addAll (items: Collection<?>): Display  
+ addLine (toAdd: Object): Display  
+ addNewLine (toAdd: Object, force: bool): Display  
+ display (): void  
+ popUp (toPop: Object): void  
+ popUp (toPop: Collection<?>): void  
+ clear (): void
```

Declaro por minha honra que este diagrama
é 100% feito sobre elementos que constituem
os grupos de projetos

Diego Alcobaça

95553