







## SwingPanel

```
- <<final>>serialVersionUID: long = 201608231505L  
- _opt: int  
- _end: boolean  
- _ins: Map<input<?>, JTextFields>  
- <<final>>_lock: Object = new Object()  
  
- <<synchronized>>sleep(n: int): void  
- <<synchronized>>await(): void  
- option(): int  
- parse(): boolean
```

## Interaction Using Swing

```
- <<final>>serialVersionUID: long = 201608231515L  
  
- message(s: String, title: String): void
```

## Display

```
- _ui: Dialog  
- _title: String = ""  
- _text: String = ""
```

```
+ add(text: String): Display  
+ addLine(text: String): Display  
+ addNewLine(text: String, force: boolean): Display  
+ getTitle(): String  
+ getText(): String  
+ display(force: boolean): void  
+ display(): void  
+ popup(toPop: Object): void
```

## <<abstract>>

## Dialog Exception

```
serialVersionUID: long = 200610291601L
```

## <<interface>> Interaction

```
+ menu(menu: Menu): void  
+ form(form: Form): void  
+ message(display: Display): void  
+ setTitle(title: String): void  
+ close(): void
```

## <<abstract>> ValidityPredicate

```
# _receiver: Receiver
```

```
+ <<abstract>>isValid(): boolean
```

## Menu

```
- _title: String  
- _commands: Command<?>[]  
- _ui: Dialog
```

```
+ title(): String  
+ size(): int  
+ entry(n: int): Command<?>  
entries(): Command<?>[]  
+ open(): void
```

## Dialog

```
- <<final>>ACTION SWING: String = "swing"  
- <<final>>ACTION TEXT: String = "text"  
+ ID: Dialog = new Dialog()  
- _subsystem: Interaction  
  
+ setTitle(title: String): void  
+ menu(menu: Menu): void  
+ form(form: Form): void  
+ message(d: Display): void  
+ close(): void
```



## Interaction Using Text

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

```
+ closeDown(): void
```

```
- <<final>> readInteger(prompt: String): int
```

```
- <<final>> readString(prompt: String): String
```

```
- <<final>> printIn(text: String): void
```

```
- <<final>> print(text: String): void
```

<<abstract>>  
Command

```
- _last: boolean
- _title: String
# _receiver: Receiver = null
# _validityPredicate: ValidityPredicate<Receiver> = null
# _form: Form
# _display: Display
```

```
+ <<final>> title(): String
+ isLast(): boolean
+ isValid(): boolean
+ <<final>> perform_command(): void
+ <<abstract>> execute(): void
```

<<abstract>>  
Input

```
- _prompt: String
- _regex: String
- _clear: boolean
# _value: Type
```

```
+ prompt(): String
+ set(value: Type): void
+ value(): Type
+ clear(): void
# dirty(): void
+ cleared(): boolean
+ <<abstract>> parse(in: String): boolean
+ regex(): String
```

Form

```
- _ui: Dialog
- _title: String
- _entries: List<Input<?>> = new ArrayList<Input<?>>()
```

```
+ title(): String
+ entries(): Collection<Input<?>>
+ entry(i: int): Input<?>
+ add(in: Input<T>): List<Input<T>>
+ addBooleanInput(label: String): Input<Boolean>
+ addStringInput(label: String): Input<String>
+ addFloatInput(label: String): Input<Float>
+ addIntegerInput(label: String): Input<Integer>
+ parse(): void
+ parse(clear: boolean): void
+ clear(): void
```

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

Mara Gomes Alves

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

Margarida Rodrigues