

## TicketOffice

```

+ reset(): void
+ save(fileName: String): void
+ load(fileName: String): void
+ importFile(dataFile: String): void
+ search(id: int, depSt: String, arrSt: String, depTime: String, arrTime: String): List<Itinerary>
+ commitItinerary(passport: int, itineraryV: int): void
+ registerPassenger(name: String): int
+ changePassengerName(id: int, name: String): void
    
```

- trains

## Service

```

- stations: Map<String, LocalTime>
- id: int
- price: double
+ service(List<String>)
+ getStations(): Map<String, LocalTime>
+ getServiceDeparture(): String
+ getServiceArrival(): String
    
```

ARRIVAL TIME

DEPART TIME

```

<<comparable>>
Comparator<Service>
    
```

```

<<comparable>>
Comparator<Service>
    
```

```

E: Class
Comparator
    
```

```

E: Class
Comparator
    
```

## TrainCompany

```

- totalPassengers: int = 0
+ registerPassenger(name: String): int
+ changePassengerName(id: int, name: String): void
+ search(id: int, depSt: String, arrSt: String, depTime: String, arrTime: String): List<Itinerary>
+ commit(passport: int, itineraryV: int): void
+ getPassenger(id: int): Passenger
+ getPassengers(): List<Passenger>
+ get_Service(id: int): Service
+ get_Services(): List<Service>
+ getServicesFromDeparture(station: String): List<Service>
+ getServiceToArrival(station: String): List<Service>
+ importFile(fileName: String): void
    
```

- passengers

```

E: Class
Comparable
    
```

```

E: Class
Comparable
    
```

- itineraries

## Itinerary

```

- orderNumber: int
- cost: double
- date: LocalDate
+ getStations(): List<String>
+ getDuration(): Duration
+ getDepartureStation(): String
+ getArrivalStation(): String
    
```

## Passenger

```

- id: int
- name: String
- totalItineraries: int
- totalPaid: double
- timesSpent: Duration
- last10Paid: double
+ Passenger(name: String)
+ commitItinerary(itineraryId: int): void
+ getItinerary(id: int): Itinerary
+ getItineraries(): List<Itinerary>
+ updateCategory(): void
    
```

- category

```

<<comparable>>
Category
- name: String
- discount: int
    
```

```

Normal
    
```

```

Special
    
```

```

Frequent
    
```

100

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

10/11/17

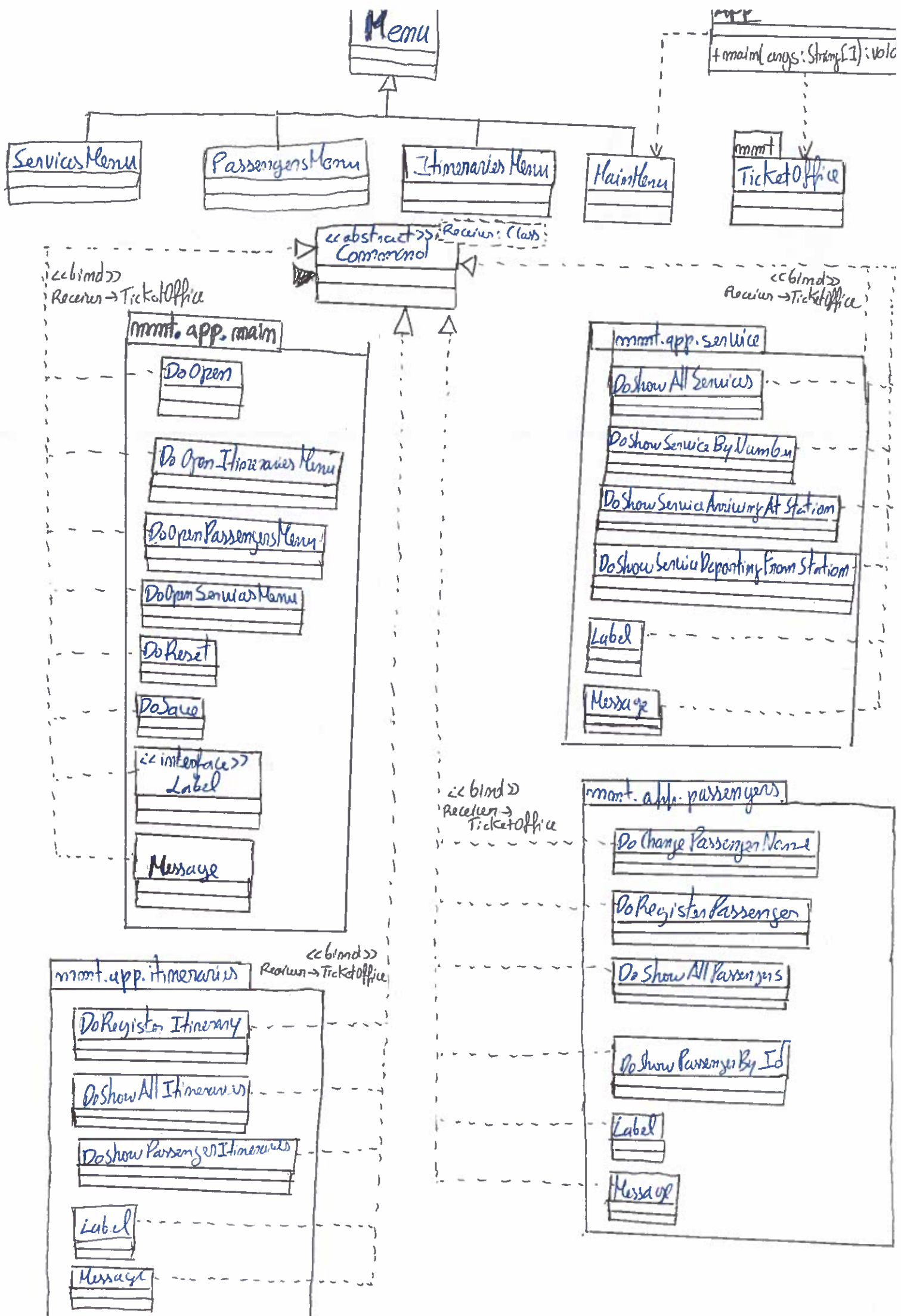
10/11/17

10/11/17

10/11/17

10/11/17

10/11/17





## Display

```
-- title: String = ""
-- text: String = ""

+ display()
+ Display(title: String)
+ Display(ui: Dialog, title: String)
+ add(text: String): Display
+ addViewLine(text: String, line: boolean): Display
+ display(line: boolean): void
+ display(): void
+ popUp(toPop: Object): void
```

-- ui

## Dialog

```
- ACTION_SWING: String = ""
- ACTION_TEST: String = ""

- Dialog()
+ Dialog(action: Interaction)
+ setTitle(title: String): void
+ menu(menu: Menu): void
+ form(form: Form): void
+ message(msg: Display): void
+ close(): void
```

+ IO

## Interaction

-- subsystem

## Command

```
+ Command()
+ Command(receiver: Receiver)

# form
```

## ValidityPredicate

```
+ ValidityPredicate()
+ ValidityPredicate(receiver: Receiver)
+ isValid(): boolean
```

# form

## Menu

```
- title: String
+ Menu(ui: Dialog, title: String, commands: Command[])
+ Menu(title: String, commands: Command[])
+ title(): String
+ size(): int
+ entry(index: Command): Command
+ entries(): Command[]
+ open(): void
```

## Input

```
+ Input()
+ InputInteger()
+ InputFloat()
+ InputInteger(prompt: String)
+ InputFloat(prompt: String)
```

-- entries

## Form

-- ui

## Input Boolean

```
+ InputBoolean(prompt: String)
+ parse(in: String): boolean
+ toString(): String
```

## Input Float

```
+ InputFloat(prompt: String)
+ parse(in: String): boolean
```

## Input Integer

```
+ InputInteger(prompt: String)
+ parse(in: String): boolean
```

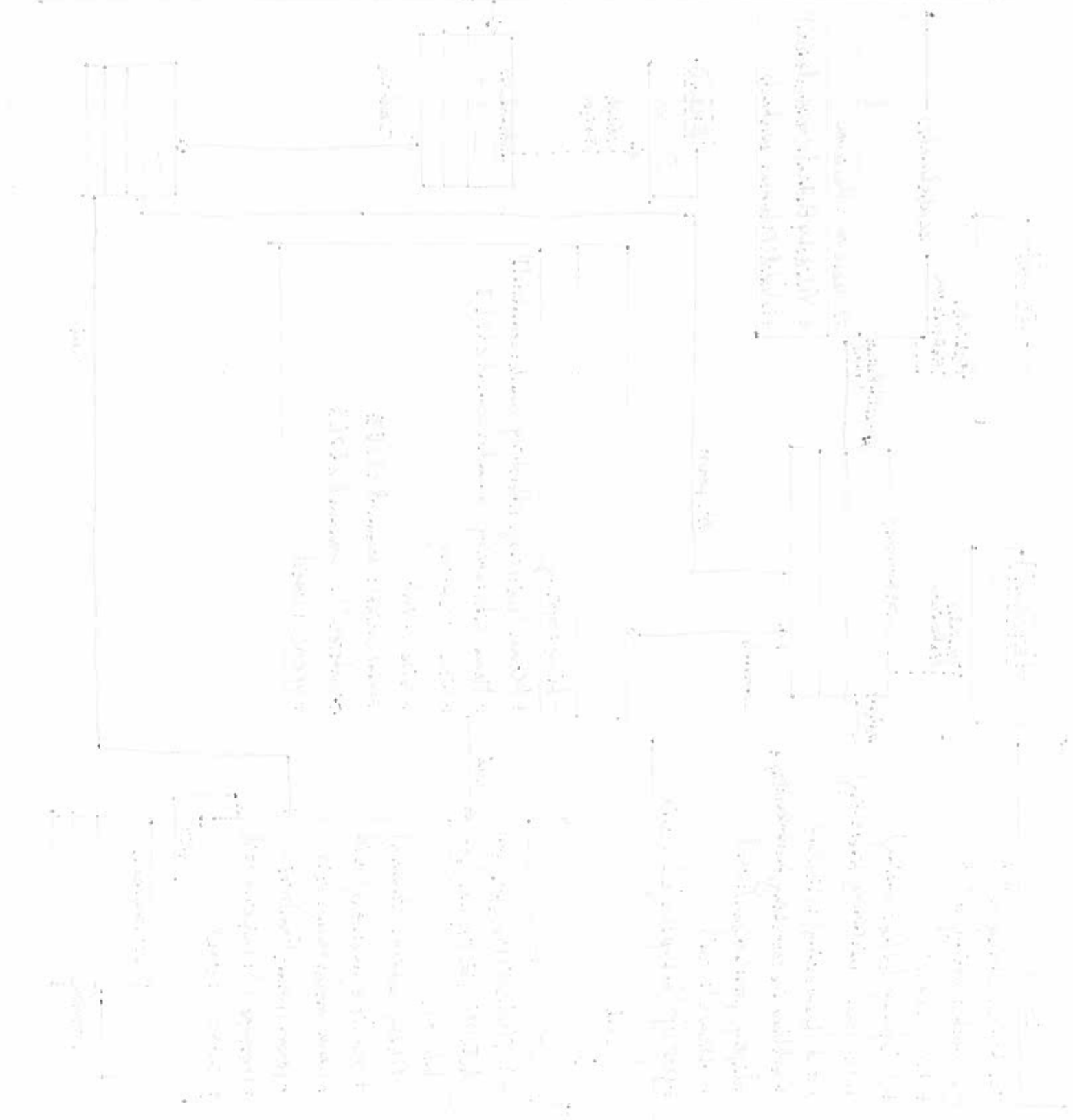
## Input None

```
+ InputNone(prompt: String)
+ parse(in: String): boolean
```

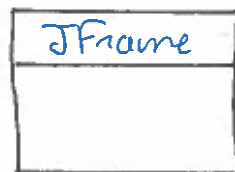
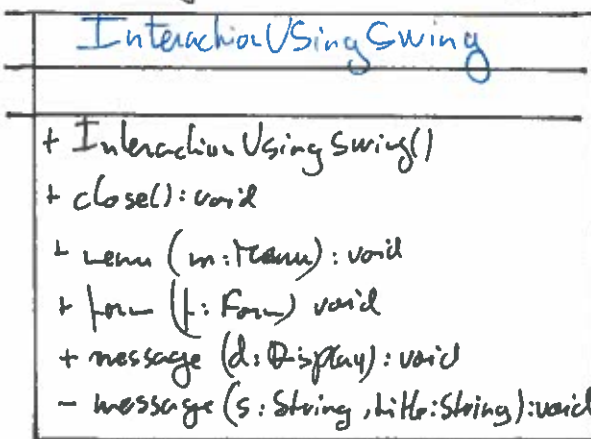
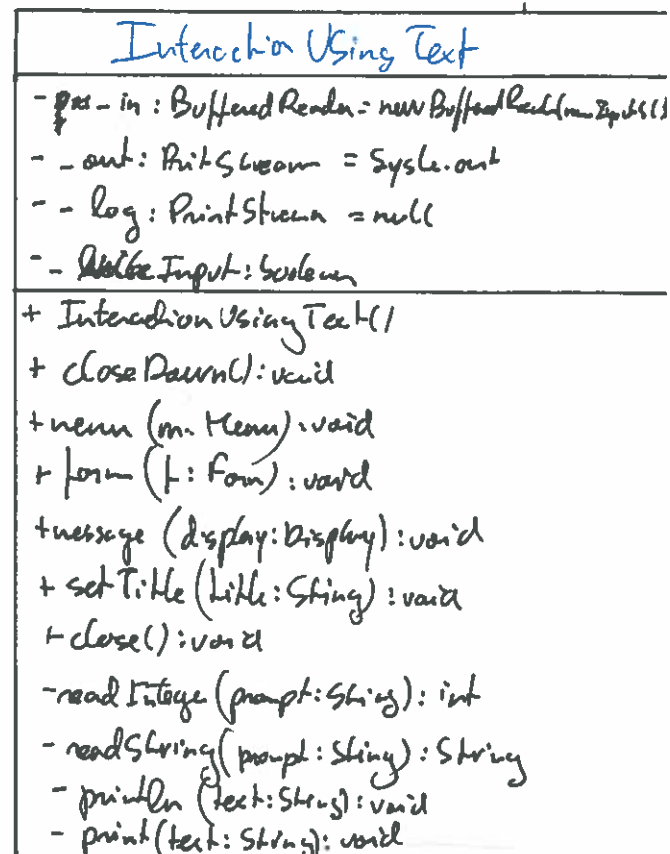
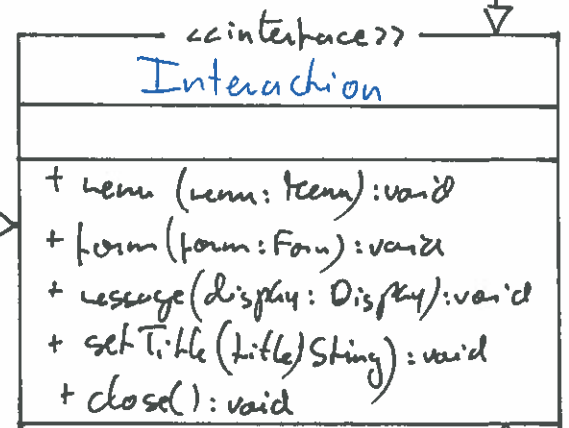
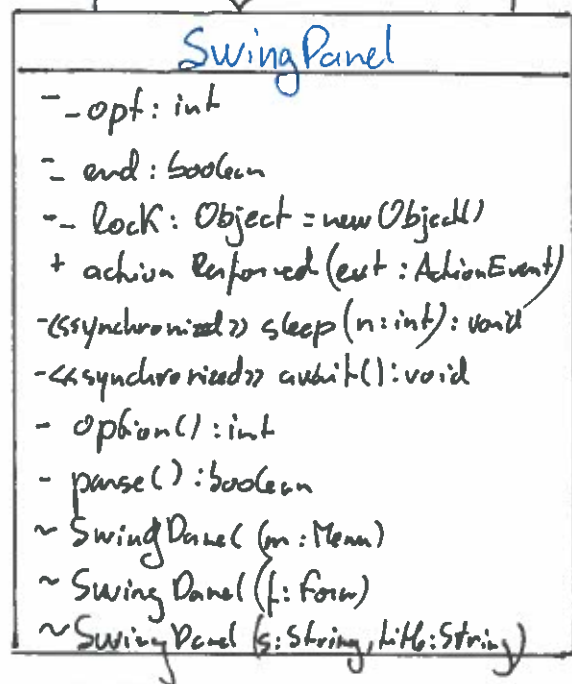
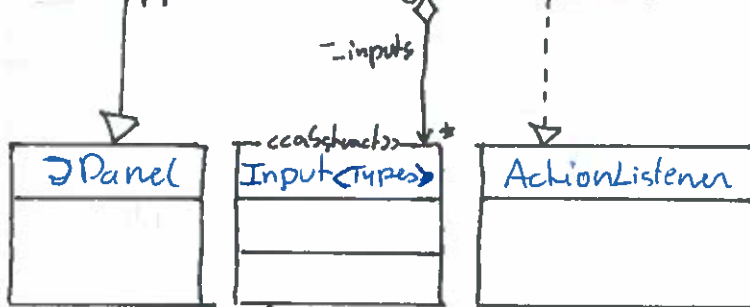
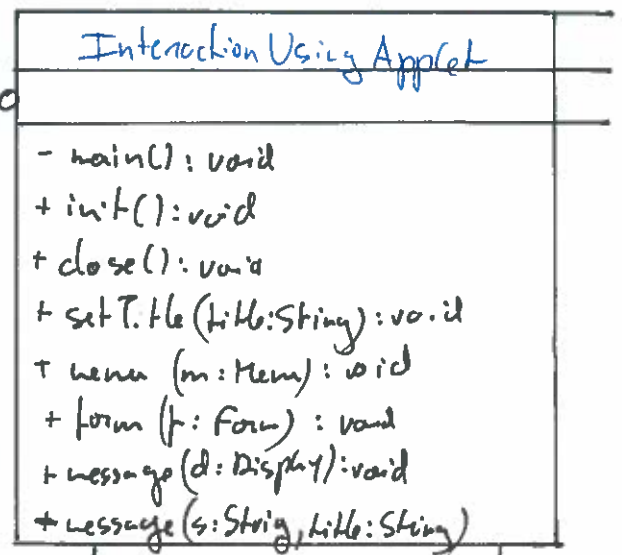
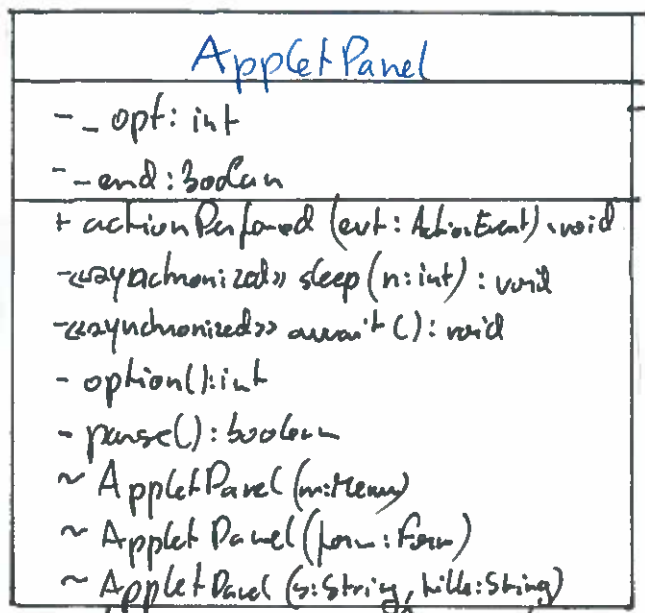
## Input String

```
+ InputString(prompt: String)
+ parse(in: String): boolean
```

<p>1. <u>Introduction</u></p> <p>The purpose of this study is to investigate the effect of the independent variable on the dependent variable.</p>	<p>2. <u>Methodology</u></p> <p>The study was conducted using a quantitative research design. The data was collected through a series of experiments.</p>
<p>3. <u>Results</u></p> <p>The results of the study show that there is a significant positive correlation between the independent variable and the dependent variable.</p>	<p>4. <u>Conclusion</u></p> <p>The study concludes that the independent variable has a significant positive effect on the dependent variable.</p>











Form	Command <Receiver>	Input <Type>
<pre> - title: String + form() + form(title: String) + form(ui: Dialog, title: String) + title(): String + entries(): Collection&lt;Input?&gt; + entry(i: Int): Input? + add(in: Input?&gt;): CT&gt; Input CT&gt; + add Boolean Input (label: String): Input Boolean? + add String Input (label: String): Input String? + add Float Input (label: String): Input Float? + add Integer Input (label: String): Input Integer? + parse(): void + parse(clean: boolean): void + clear(): void </pre>	<pre> - last: boolean - title: String # - receiver: Receiver = null + Command(last: boolean, title: String) + Command(title: String, receiver: Receiver) + Command(last: boolean, title: String, receiver: Receiver) + Command(title: String, receiver: Receiver, predicate: ValidationPredicate) + isLast(): boolean + isValid(): boolean + performCommand(): void + &lt;&lt;abstract&gt;&gt; execute(): void + title(): String </pre>	<pre> - &lt;&lt;abstract&gt;&gt; Input &lt;Type&gt; - prompt: String - regex: String - clean: boolean # - value: Type + Prompt(): String # Input() # Input(prompt: String, regex: String) + set(value: Type): void + value(): Type + clear(): void # dirt(): void + clear(): boolean + &lt;&lt;abstract&gt;&gt; parse(in: String): boolean + regex(): String + toString(): String </pre>

Optamos por não incluir as classes Constants, ErrorMessage, Messages e PropertyNames, por serem apenas final, static Strings. Também não incluímos a DialogException, CompositePrintStream e RuntimeEOFException por não acrescentarem nada de relevante ao UML.



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

Youn Daniel Silva 86445



Declaro por minha honra que este diagrama foi realizado apenas pelos membros que constituem o grupo do projeto.

Francisco do Carmo Sousa 86416

Justicia per omnia seculorum  
in hac vita et in futura  
ex parte de iudicio.

Procurator de iure datus 1544