```
                          ┌─────────────────────────────┐
                          │            Light            │
                          ├─────────────────────────────┤
                          │ - MAX_BRIGHTNESS: int        │
                          │ - MAX_COLOR: Color           │
                          │                             │
                          │ - lightColor: Color          │
                          │ - panelWidth: int            │
                          │ - panelHeight: int           │
          0..1            │ - numLights: int             │
                          │ - widthPad: int              │
                          │ - heightPad: int             │
                          │ - gap: int                   │
                          │ - circleSize: int            │
                          │ - radius: int                │
                          ├─────────────────────────────┤
                          │ + Light (int, int, int)      │
```

Extends

The Appliance superclass could include a
light, fan, radiator, window, curtains, A/C,
fire alarm, etc.

```
                                    ┌─────────────────────────────┐
                                    │          Appliance          │
                                    ├─────────────────────────────┤
                                    │ + panel: AppliancePanel      │
```

ss of >

The ApplianceApp subclasses contain the main method - the program will be either a FanApp with Fan components or a LightApp with Light components depending on which ApplianceApp subclass is built.

These panels initialise the relevant

FanApp

BrightnessFormatException

+ BrightnessFormatException (String

Extends

DelayFormatException

+ DelayFormatException (String)

Extends

UserInputException

input: String
inputType: String
rule: String

+ UserInputException(String)
+ toString(): String

Extends

PortFormatException

+ PortFormatException (String)
+ PortFormatException (int)

UML Class Diagram

Fan
- triWidth: int
- arcWidth: int
- centerX: int
- centerY: int
- angle: double
- fanSpeed: int

- + Fan(int, int)
- + update()
- - addRad(double, double)
- + draw(Graphics)
- + getFanSpeed(): int
- + setFanSpeed(int)

(top, partially visible)
- + draw(Graphics)
- + setLightColor (int)
- + getCurrentBrightness(): int

- + setPanel (AppliancePanel)
- + *draw (Graphics)*

adjusts the brightne...

Extends

creates and displays>

«interface»
ActionListener

<<implements>>

1

AppliancePanel
- # applianceInstance: Appliance
- # backgroundColor: Color

- + AppliancePanel (int, int, Color)
- + getAppliance(): Appliance
- + paintComponent (Graphics)

0..1

Extends

0..1

The client accepts messages from the server and passes the new temperature or light readings to the Speed or Brightness Panel (respectively).

BrightnessPanel
- lightInstance: Light
- lastValidInput: String
- app: LightApp

- + BrightnessPanel (Light, String, String, Str
- + setButtonActions()
- + actionPerformed(ActionEvent)
- + convertReading (double)
- + setBrightness (int)
- + setLightApp (LightApp)

Extends

<<implements>>

< passes on messages from the server to

java.swing.JPanel

< creates

Extends

< creates

SpeedPanel
- fanInstance: Fan
- app: FanApp
- lastValidValue: String

- + SpeedPanel (Fan, String, String, String)
- + setButtonActions()
- + actionPerformed(ActionEvent)
- + convertReading (double)
- + getFanSpeedValue(): String
- + setFanAppInstance (FanApp)
- + setFanInstance (Fan)

Extends

ValueButtonPanel
- # label: JLabel
- # textField: JTextField
- # button: JButton

- + ValueButtonPanel(String, String, String)
- # *setButtonActions()*

0..1

«interface»
Runnable

<<implements>>

< closes on exit

adjusts the speed of >

<<throws>>

<<throws>>

Extends

java.lang.Exception

The client accepts messages from the server and passes the new temperature or light readings to the Speed or Brightness Panel (respectively).

0..1

LightClient
- brightnessPanelInstance: BrightnessPanel
- app: LightApp

- + LightClient (AbstractClientConnectPanel, int, ApplianceApp)
- + run()

Extends

AbstractClient
- - IP: String

- - port: int
- # cPanel: AbstractClientConnectPanel
- # reader: ObjectInputStream
- # writer: ObjectOutputStream
- # socket: Socket
- # clientType: String
- # stopClient: boolean
- # connectionsClosed: boolean

- + AbstractClient (AbstractClientConnectPanel, int)
- + run()
- + initialiseClient()
- + displayMessage()
- + closeAll()
- # sendMessageToServer (String)
- + setStopClient (boolean)

1

accepts &

0..1

< passes on messages from the server to

FanClient
- speedPanelInstance: SpeedPanel
- app: FanApp

- + FanClient (AbstractClientConnectPanel, int, ApplianceApp)
- + run()

Extends

0..1

Extends

<<throws>>

Appliance and display it within their panel.

LightPanel
___
+ LightPanel (int, int)

ends

FanPanel
___
- DEFAULT_DELAY_TIME: int

- timer: Timer
___
+ FanPanel(int, int)
+ setTimer(int)
+ actionPerformed(ActionEvent)

Extends

FanApp
___
- fanUiInstance: FanApp

- fanPanel: FanPanel
- speedPanel: SpeedPanel
- fanInstance: Fan
___
+ main (String[])
+ getUiInstance(): FanApp

+ FanApp()
# draw()
# initialiseCPanel()
# initialiseValueButtonPanel()
# initialiseAppliancePanel()
+ getSpeedPanelInstance(): SpeedPanel

LightApp
___
- lightUiInstance: LightApp

- lightPanel: LightPanel
- bPanel: BrightnessPanel
- lightInstance: Light
___
+ main (String[])
+ getUiInstance(): LightApp

+ LightApp()
# draw()
# initialiseCPanel()
# initialiseValueButtonPanel()
# initialiseAppliancePanel()
+ getBrightnessPanelInstance():
BrightnessPanel

Extends

ApplianceApp
___
+ client: AbstractClient
# cPanel: AbstractClientConnectPanel
# port: int
# ip: String
# clientType: String
___
+ go()
+ displayMessage (String)
+ showUserErrorDialog (String, String)

 # draw ()
 # initialiseCPanel()
 # initialiseValueButtonPanel()
 # initialiseAppliancePanel()

Extends

1

1

1

1

The ApplianceApp initialises the panels that appear on the GUI, including the ClientConnectPanel which initialises the client.

Extends

java.swing.JFrame

creates and sends
messages to >

ScrollingTextBox
___
- textBox: JTextArea
- scrollPane: JScrollPane
- scrollBar: JScrollBar
___
+ getScrollPane(): JScrollPane
+ displayMessage(String)
+ scrollToBottom()

1

creates and
sends messages to >

0..1

< creates,
starts/stops,
displays messages from

1

AbstractClientConnectPanel
___
DEFAULT_PORT: String
MIN_PORT_NUM: int
MAX_PORT_NUM: int

lastReading: String
clientOutput: ScrollingTextBox
app: ApplienceApp
containerPanel: JPanel
client: AbstractClient
port: int
stopButton: JButton
___
+ AbstractClientConnectPanel (String, String)
# setButtonActions()
+ actionPerformed (ActionEvent)
# stopClient()
+ displayMessage (String)
+ showUserErrorDialog (String, String)
+ getContainerPanel (): JPanel
+ setApp (ApplianceApp)
+ getPortNumber(): int
+ resetButtons()

 # startClient()

FanClientConnectPanel
___
+ FanClientConnectPanel(String, String)
+ startClient()

Extends

LightClientConnectPanel
___
+ LightClientConnectPanel(String, String)
+ startClient()

Extends