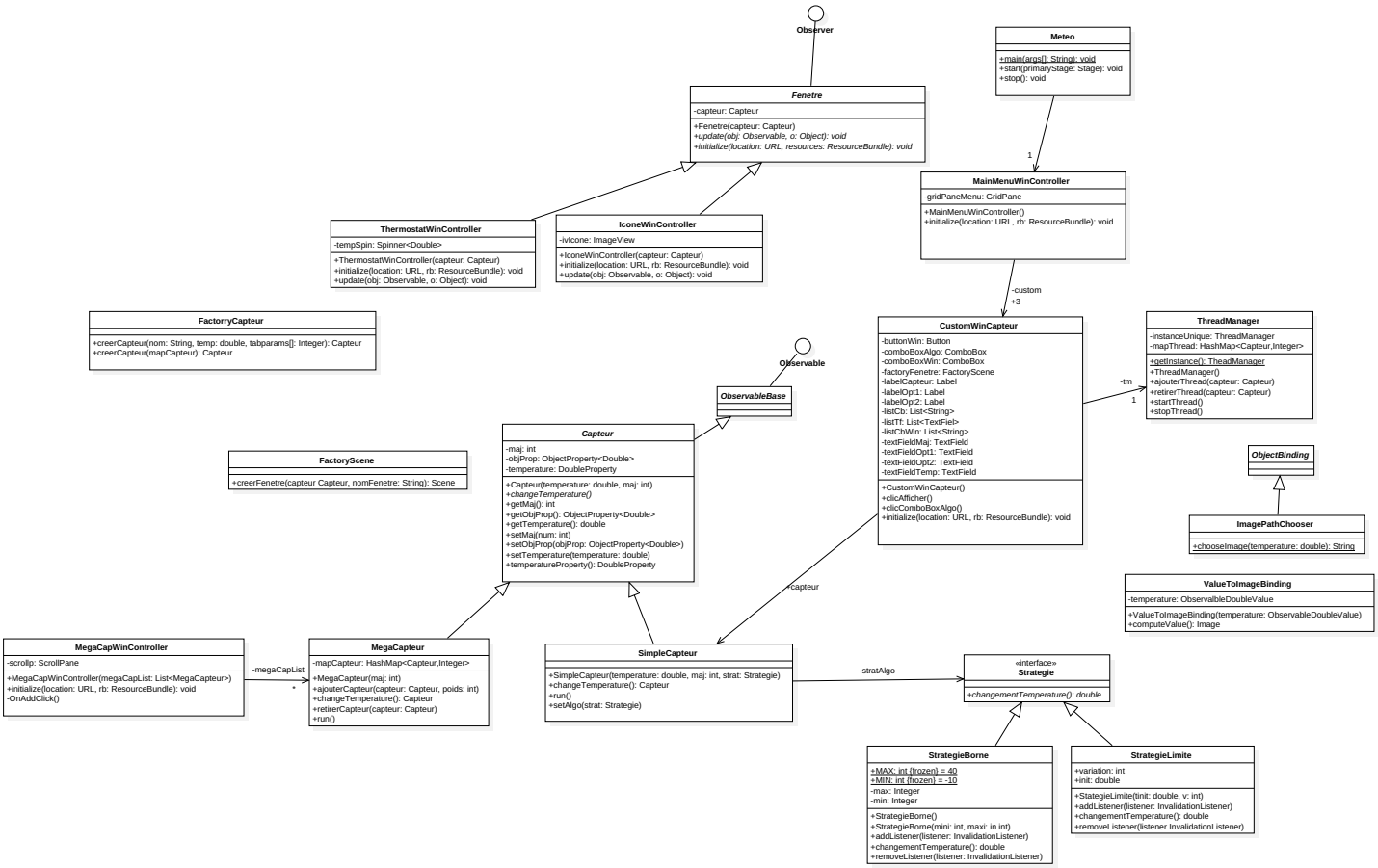
Model::Main


Observer
Meteo
+main(args[]: String): void
+start(primaryStage: Stage): void
+stop(): void
Fenetre
-capteur: Capteur
+Fenetre(capteur: Capteur)
+update(obj: Observable, o: Object): void
+initialize(location: URL, resources: ResourceBundle): void
1
MainMenuWinController
-gridPaneMenu: GridPane
+MainMenuWinController()
+initialize(location: URL, rb: ResourceBundle): void
ThermostatWinController
-tempSpin: Spinner<Double>
+ThermostatWinController(capteur: Capteur)
+initialize(location: URL, rb: ResourceBundle): void
+update(obj: Observable, o: Object): void
IconeWinController
-lvIcone: ImageView
+IconeWinController(capteur: Capteur)
+initialize(location: URL, rb: ResourceBundle): void
+update(obj: Observable, o: Object): void
-custom
+3
FactorryCapteur
+creerCapteur(nom: String, temp: double, tabparams[]: Integer): Capteur
+creerCapteur(mapCapteur): Capteur
CustomWinCapteur
-buttonWin: Button
-comboBoxAlgo: ComboBox
-comboBoxWin: ComboBox
-factoryFenetre: FactoryScene
-labelCapteur: Label
-labelOpt1: Label
-labelOpt2: Label
-listCb: List<String>
-listTf: List<TextFiel>
-listCbWin: List<String>
-textFieldMaj: TextField
-textFieldOpt1: TextField
-textFieldOpt2: TextField
-textFieldTemp: TextField
+CustomWinCapteur()
+clicAfficher()
+clicComboBoxAlgo()
+initialize(location: URL, rb: ResourceBundle): void
ThreadManager
-instanceUnique: ThreadManager
-mapThread: HashMap<Capteur,Integer>
+getInstance(): TheadManager
+ThreadManager()
+ajouterThread(capteur: Capteur)
+retirerThread(capteur: Capteur)
+startThread()
+stopThread()
-tm
1
Observable
ObservableBase
Capteur
-maj: int
-objProp: ObjectProperty<Double>
-temperature: DoubleProperty
+Capteur(temperature: double, maj: int)
+changeTemperature()
+getMaj(): int
+getObjProp(): ObjectProperty<Double>
+getTemperature(): double
+setMaj(num: int)
+setObjProp(objProp: ObjectProperty<Double>)
+setTemperature(temperature: double)
+temperatureProperty(): DoubleProperty
FactoryScene
+creerFenetre(capteur Capteur, nomFenetre: String): Scene
ObjectBinding
ImagePathChooser
+chooseImage(temperature: double): String
-capteur
ValueToImageBinding
-temperature: ObservalbleDoubleValue
+ValueToImageBinding(temperature: ObservableDoubleValue)
+computeValue(): Image
MegaCapWinController
-scrollp: ScrollPane
+MegaCapWinController(megaCapList: List<MegaCapteur>)
+initialize(location: URL, rb: ResourceBundle): void
-OnAddClick()
-megaCapList
*
MegaCapteur
-mapCapteur: HashMap<Capteur,Integer>
+MegaCapteur(maj: int)
+ajouterCapteur(capteur: Capteur, poids: int)
+changeTemperature(): Capteur
+retirerCapteur(capteur: Capteur)
+run()
SimpleCapteur
+SimpleCapteur(temperature: double, maj: int, strat: Strategie)
+changeTemperature(): Capteur
+run()
+setAlgo(strat: Strategie)
-stratAlgo
«interface»
Strategie
+changementTemperature(): double
StrategieBorne
+MAX: int (frozen) = 40
+MIN: int (frozen) = -10
-max: Integer
-min: Integer
+StrategieBorne()
+StrategieBorne(mini: int, maxi: in int)
+addListener(listener: InvalidationListener)
+changementTemperature(): double
+removeListener(listener: InvalidationListener)
StrategieLimite
+variation: int
+init: double
+StategieLimite(tinit: double, v: int)
+addListener(listener: InvalidationListener)
+changementTemperature(): double
+removeListener(listener InvalidationListener)