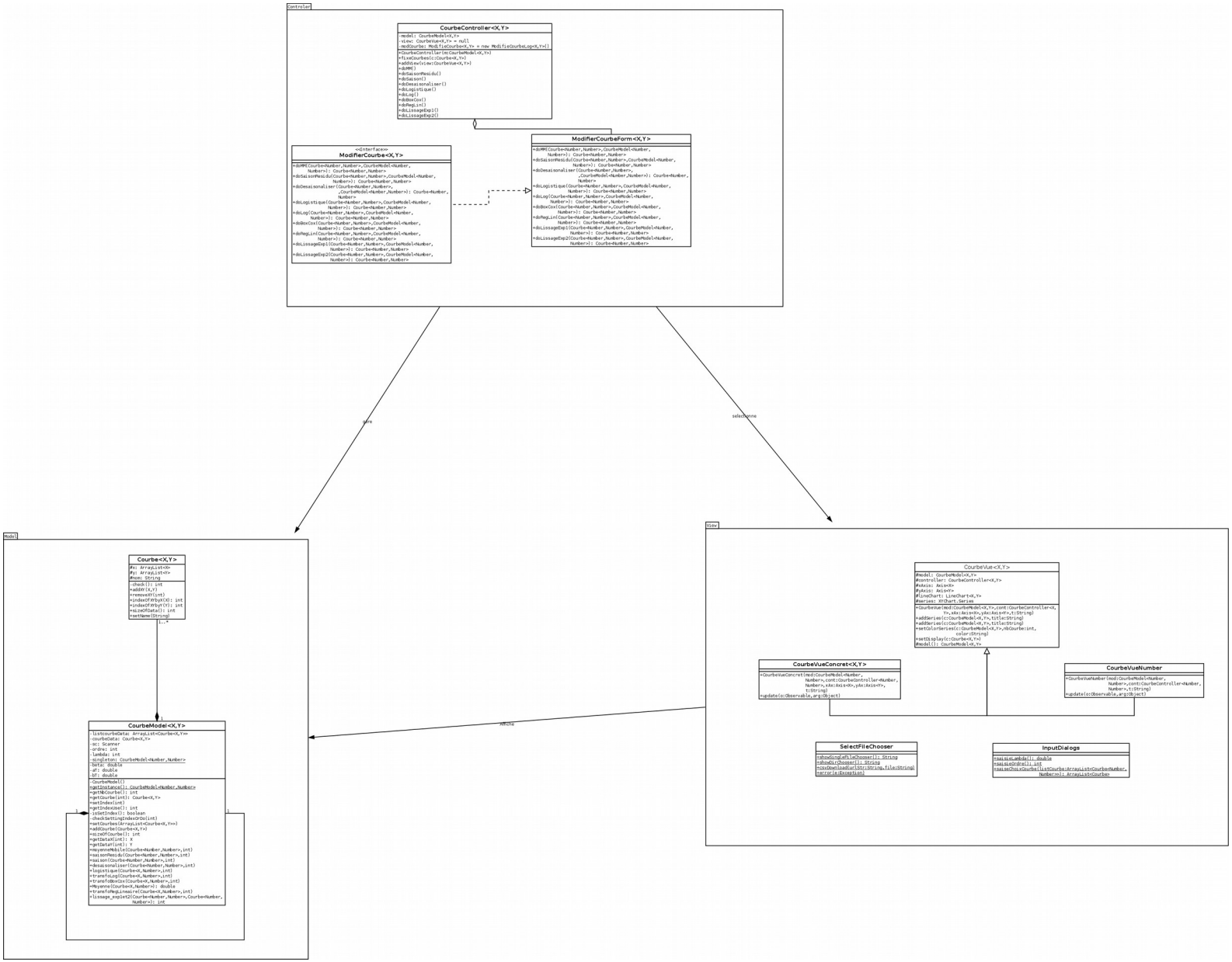
Controler
CourbeController<X,Y>
-model: CourbeModel<X,Y>
-view: CourbeVue<X,Y> = null
#modCourbe: ModifieCourbe<X,Y> = new ModifieCourbeLog<X,Y>()
+CourbeController(m:CourbeModel<X,Y>)
+fixeCourbes(c:Courbe<X,Y>)
+addView(v:CourbeVue<X,Y>)
+dsMMC()
+dsSaisonResidu()
+dsSaison()
+dsDesaisonaliser()
+dsLogistique()
+dsLog()
+dsBoxCox()
+dsRegLin()
+dsLissageExp1()
+dsLissageExp2()
<<Interface>>
ModifierCourbe<X,Y>
+dsMMC(Courbe<Number,Number>,CourbeModel<Number, Number>): Courbe<Number,Number>
+dsSaisonResidu(Courbe<Number,Number>,CourbeModel<Number, Number>): Courbe<Number,Number>
+dsDesaisonaliser(Courbe<Number,Number>, ,CourbeModel<Number,Number>): Courbe<Number, Number>
+dsLogistique(Courbe<Number,Number>,CourbeModel<Number, Number>): Courbe<Number,Number>
+dsLog(Courbe<Number,Number>,CourbeModel<Number, Number>): Courbe<Number,Number>
+dsBoxCox(Courbe<Number,Number>,CourbeModel<Number, Number>): Courbe<Number,Number>
+dsRegLin(Courbe<Number,Number>,CourbeModel<Number, Number>): Courbe<Number,Number>
+dsLissageExp1(Courbe<Number,Number>,CourbeModel<Number, Number>): Courbe<Number,Number>
+dsLissageExp2(Courbe<Number,Number>,CourbeModel<Number, Number>): Courbe<Number,Number>
ModifierCourbeForm<X,Y>
+dsMMC(Courbe<Number,Number>,CourbeModel<Number, Number>): Courbe<Number,Number>
+dsSaisonResidu(Courbe<Number,Number>,CourbeModel<Number, Number>): Courbe<Number,Number>
+dsDesaisonaliser(Courbe<Number,Number>, ,CourbeModel<Number,Number>): Courbe<Number, Number>
+dsLogistique(Courbe<Number,Number>,CourbeModel<Number, Number>): Courbe<Number,Number>
+dsLog(Courbe<Number,Number>,CourbeModel<Number, Number>): Courbe<Number,Number>
+dsBoxCox(Courbe<Number,Number>,CourbeModel<Number, Number>): Courbe<Number,Number>
+dsRegLin(Courbe<Number,Number>,CourbeModel<Number, Number>): Courbe<Number,Number>
+dsLissageExp1(Courbe<Number,Number>,CourbeModel<Number, Number>): Courbe<Number,Number>
+dsLissageExp2(Courbe<Number,Number>,CourbeModel<Number, Number>): Courbe<Number,Number>
sera
selectionne
Model
Courbe<X,Y>
#x: ArrayList<X>
#y: ArrayList<Y>
#nom: String
+check(): int
+addXY(x,y)
+removeXY(int)
+indexOfXY(X): int
+indexOfY(Y): int
+sizeOfData(): int
+setName(String)
1..*
CourbeModel<X,Y>
-listcourbeData: ArrayList<Courbe<X,Y>>
-courbeData: Courbe<X,Y>
-sc: Scanner
-ordre: int
-lambda: int
-singleton: CourbeModel<Number,Number>
-beta: double
-alf: double
-bf: double
-CourbeModel()
+getInstance(): CourbeModel<Number,Number>
+getNbCourbe(): int
+getCourbe(int): Courbe<X,Y>
+setIndex(int)
+getIndexUse(): int
-isSetIndex(): boolean
-checkSettingIndexOrDo(int)
+addCourbes(ArrayList<Courbe<X,Y>>)
+addCourbe(Courbe<X,Y>)
+sizeOfCourbe(): int
+getDataX(int): X
+getDataY(int): Y
+moyenneMobile(Courbe<Number,Number>,int)
+saisonResidu(Courbe<Number,Number>,int)
+saison(Courbe<Number,Number>,int)
+desaisonaliser(Courbe<Number,Number>,int)
+logistique(Courbe<X,Number>,int)
+transfoLog(Courbe<X,Number>,int)
+transfoBoxCox(Courbe<X,Number>,int)
+Moyenne(Courbe<X,Number>): double
+transfoRegLineaire(Courbe<X,Number>,int)
+lissage_expint2(Courbe<Number,Number>,Courbe<Number, Number>): int
1
1
View
CourbeVue<X,Y>
#model: CourbeModel<X,Y>
#controller: CourbeController<X,Y>
#xAxis: Axis<X>
#yAxis: Axis<Y>
#lineChart: LineChart<X,Y>
#series: XYChart.Series
+CourbeVue(mod:CourbeModel<X,Y>,cont:CourbeController<X, Y>,xAx:Axis<X>,yAx:Axis<Y>,t:String)
+addSeries(c:CourbeModel<X,Y>,title:String)
+addSeries(c:CourbeModel<X,Y>,title:String)
+setColorSeries(c:CourbeModel<X,Y>,nbCourbe:int, color:String)
+setDisplay(c:Courbe<X,Y>)
#model(): CourbeModel<X,Y>
CourbeVueConcret<X,Y>
+CourbeVueConcret(mod:CourbeModel<Number, Number>,cont:CourbeController<Number, Number>,xAx:Axis<X>,yAx:Axis<Y>, t:String)
+update(o:Observable,arg:Object)
CourbeVueNumber
+CourbeVueNumber(mod:CourbeModel<Number, Number>,cont:CourbeController<Number, Number>,t:String)
+update(o:Observable,arg:Object)
SelectFileChooser
+showDialogFileChooser(): String
+showDirChooser(): String
+saveDownloadDir(StrString,fileStString)
+error(e:Exception)
InputDialogs
+saisieLambda(): double
+saisieOrdre(): int
+saisieChoixCourbe(listCourbe:ArrayList<Courbe<Number, Number>>): ArrayList<Courbe>
affecte