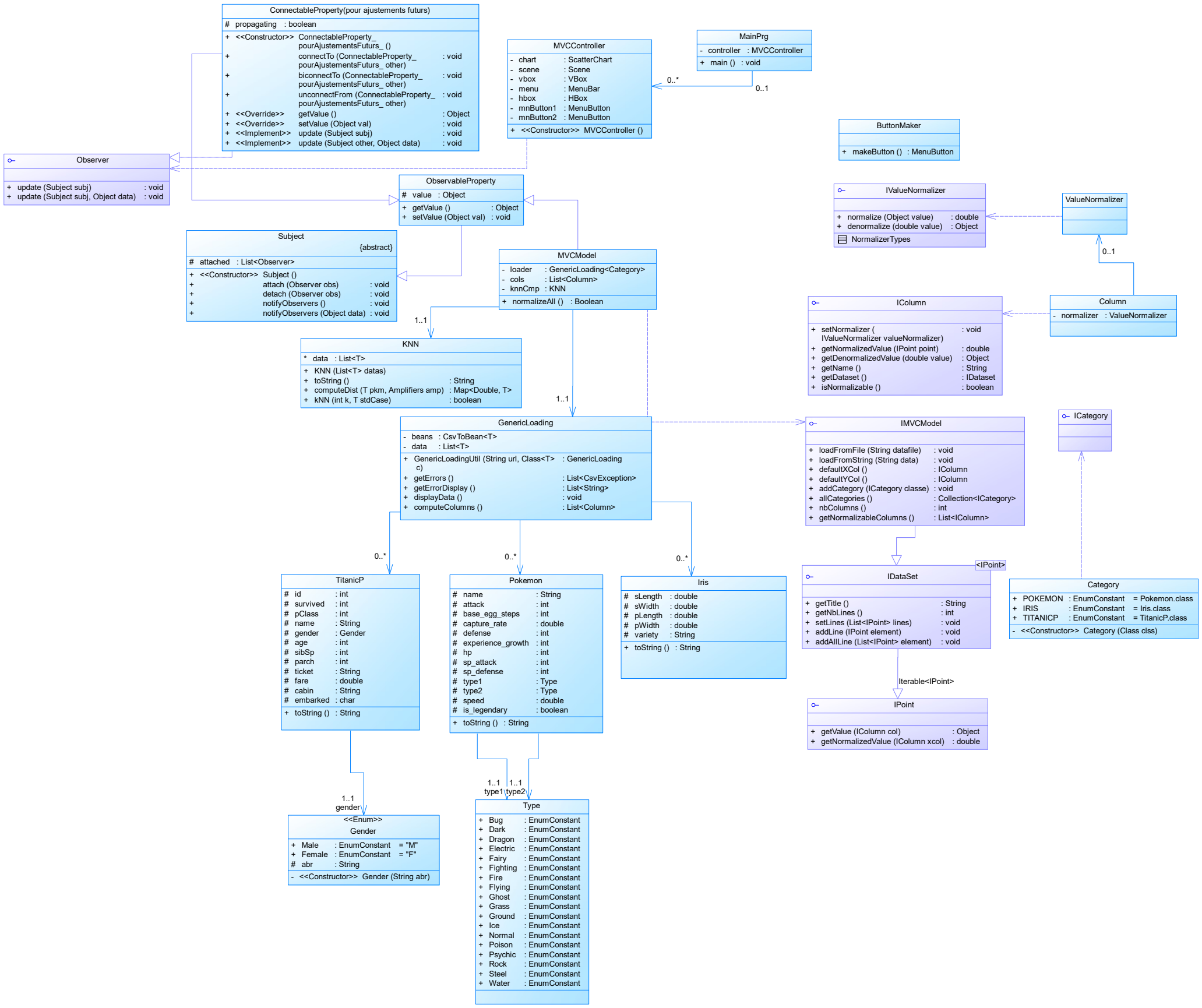
ConnectableProperty(pour ajustements futurs)
# propagating : boolean
+ <<Constructor>> ConnectableProperty_ pourAjustementsFuturs_ ()
+ connectTo (ConnectableProperty_ pourAjustementsFuturs_ other) : void
+ biconnectTo (ConnectableProperty_ pourAjustementsFuturs_ other) : void
+ unconnectFrom (ConnectableProperty_ pourAjustementsFuturs_ other) : void
+ <<Override>> getValue () : Object
+ <<Override>> setValue (Object val) : void
+ <<Implement>> update (Subject subj) : void
+ <<Implement>> update (Subject other, Object data) : void
MVCController
- chart : ScatterChart
- scene : Scene
- vbox : VBox
- menu : MenuBar
- hbox : HBox
- mnButton1 : MenuButton
- mnButton2 : MenuButton
+ <<Constructor>> MVCController ()
MainPrg
- controller : MVCController
+ main () : void
0..*
0..1
ButtonMaker
+ makeButton () : MenuButton
Observer
+ update (Subject subj) : void
+ update (Subject subj, Object data) : void
ObservableProperty
value : Object
+ getValue () : Object
+ setValue (Object val) : void
IValueNormalizer
+ normalize (Object value) : double
+ denormalize (double value) : Object
NormalizerTypes
ValueNormalizer
0..1
Subject
{abstract}
attached : List<Observer>
+ <<Constructor>> Subject ()
+ attach (Observer obs) : void
+ detach (Observer obs) : void
+ notifyObservers () : void
+ notifyObservers (Object data) : void
MVCModel
- loader : GenericLoading<Category>
- cols : List<Column>
- knnCmp : KNN
+ normalizeAll () : Boolean
1..1
IColumn
+ setNormalizer (IValueNormalizer valueNormalizer) : void
+ getNormalizedValue (IPoint point) : double
+ getDenormalizedValue (double value) : Object
+ getName () : String
+ getDataset () : IDataset
+ isNormalizable () : boolean
Column
- normalizer : ValueNormalizer
KNN
* data : List<T>
+ KNN (List<T> datas)
+ toString () : String
+ computeDist (T pkm, Amplifiers amp) : Map<Double, T>
+ kNN (int k, T stdCase) : boolean
1..1
GenericLoading
- beans : CsvToBean<T>
- data : List<T>
+ GenericLoadingUtil (String url, Class<T> c) : GenericLoading
+ getErrors () : List<CsvException>
+ getErrorDisplay () : List<String>
+ displayData () : void
+ computeColumns () : List<Column>
IMVCModel
+ loadFromFile (String datafile) : void
+ loadFromString (String data) : void
+ defaultXCol () : IColumn
+ defaultYCol () : IColumn
+ addCategory (ICategory classe) : void
+ allCategories () : Collection<ICategory>
+ nbColumns () : int
+ getNormalizableColumns () : List<IColumn>
ICategory
0..*
0..*
0..*
TitanicP
id : int
survived : int
pClass : int
name : String
gender : Gender
age : int
sibSp : int
parch : int
ticket : String
fare : double
cabin : String
embarked : char
+ toString () : String
Pokemon
name : String
attack : int
base_egg_steps : int
capture_rate : double
defense : int
experience_growth : int
hp : int
sp_attack : int
sp_defense : int
type1 : Type
type2 : Type
speed : double
is_legendary : boolean
+ toString () : String
Iris
sLength : double
sWidth : double
pLength : double
pWidth : double
variety : String
+ toString () : String
<IPoint>
IDataSet
+ getTitle () : String
+ getNbLines () : int
+ setLines (List<IPoint> lines) : void
+ addLine (IPoint element) : void
+ addAllLine (List<IPoint> element) : void
Category
+ POKEMON : EnumConstant = Pokemon.class
+ IRIS : EnumConstant = Iris.class
+ TITANICP : EnumConstant = TitanicP.class
- <<Constructor>> Category (Class clss)
Iterable<IPoint>
IPoint
+ getValue (IColumn col) : Object
+ getNormalizedValue (IColumn xcol) : double
1..1
gender
<<Enum>>
Gender
+ Male : EnumConstant = "M"
+ Female : EnumConstant = "F"
# abr : String
- <<Constructor>> Gender (String abr)
1..1
type1
1..1
type2
Type
+ Bug : EnumConstant
+ Dark : EnumConstant
+ Dragon : EnumConstant
+ Electric : EnumConstant
+ Fairy : EnumConstant
+ Fighting : EnumConstant
+ Fire : EnumConstant
+ Flying : EnumConstant
+ Ghost : EnumConstant
+ Grass : EnumConstant
+ Ground : EnumConstant
+ Ice : EnumConstant
+ Normal : EnumConstant
+ Poison : EnumConstant
+ Psychic : EnumConstant
+ Rock : EnumConstant
+ Steel : EnumConstant
+ Water : EnumConstant