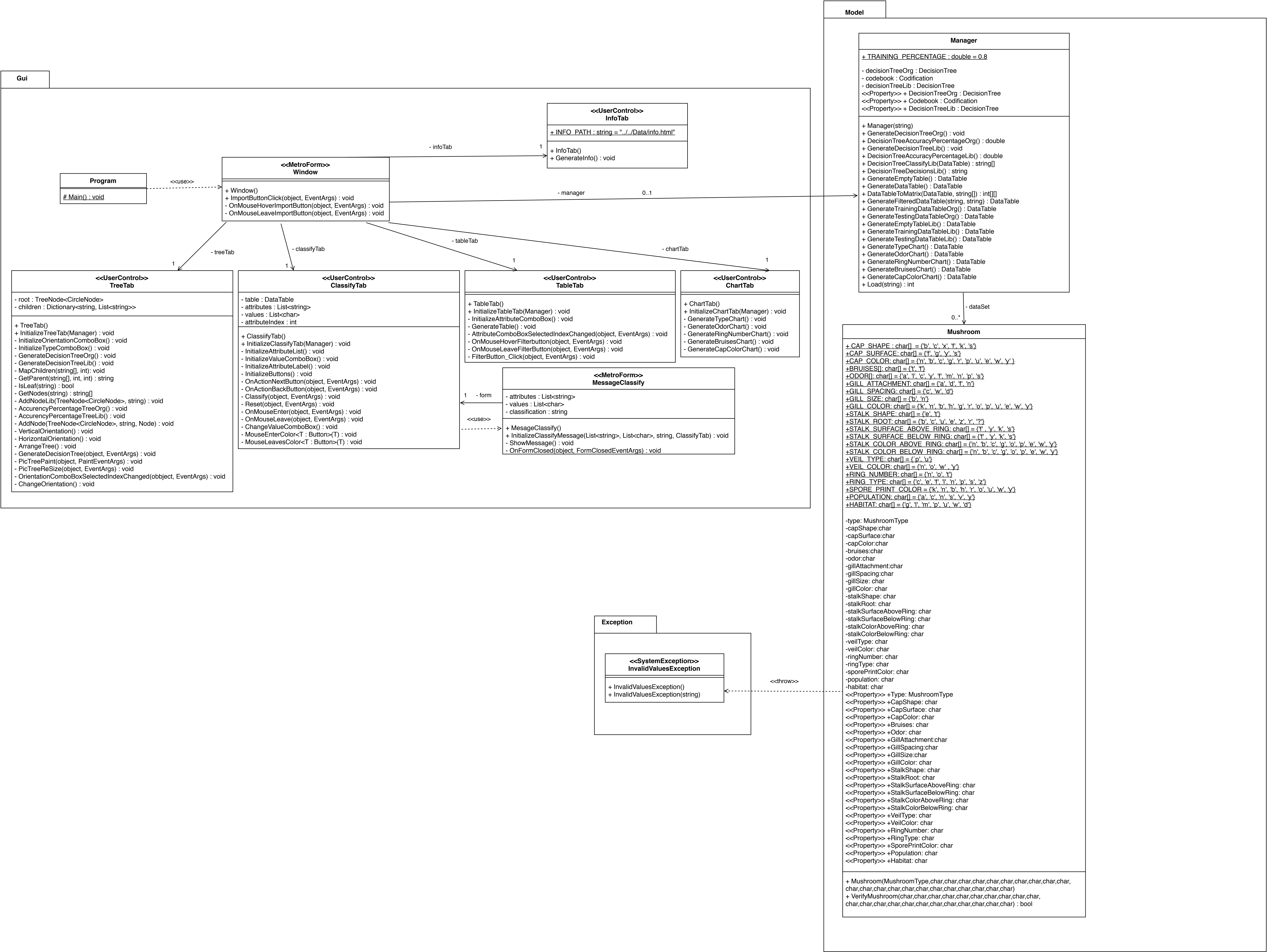
Gui
Program
# Main() : void
<<use>>
<<MetroForm>>
Window
+ Window()
+ ImportButtonClick(object, EventArgs) : void
- OnMouseHoverImportButton(object, EventArgs) : void
- OnMouseLeaveImportButton(object, EventArgs) : void
- infoTab
1
<<UserControl>>
InfoTab
+ INFO_PATH : string = "../../Data/info.html"
+ InfoTab()
+ GenerateInfo() : void
- manager
0..1
- treeTab
1
- classifyTab
1
- tableTab
1
- chartTab
1
<<UserControl>>
TreeTab
- root : TreeNode<CircleNode>
- children : Dictionary<string, List<string>>
+ TreeTab()
+ InitializeTreeTab(Manager) : void
- InitializeOrientationComboBox() : void
- InitializeTypeComboBox() : void
- GenerateDecisionTreeOrg() : void
- GenerateDecisionTreeLib() : void
- MapChildren(string[], int): void
- GetParent(string[], int, int) : string
- IsLeaf(string) : bool
- GetNodes(string) : string[]
- AddNodeLib(TreeNode<CircleNode>, string) : void
- AccurencyPercentageTreeOrg() : void
- AccurencyPercentageTreeLib() : void
- AddNode(TreeNode<CircleNode>, string, Node) : void
- VerticalOrientation() : void
- HorizontalOrientation() : void
- ArrangeTree() : void
- GenerateDecisionTree(object, EventArgs) : void
- PicTreePaint(object, PaintEventArgs) : void
- PicTreeReSize(object, EventArgs) : void
- OrientationComboBoxSelectedIndexChanged(obbject, EventArgs) : void
- ChangeOrientation() : void
<<UserControl>>
ClassifyTab
- table : DataTable
- attributes : List<string>
- values : List<char>
- attributeIndex : int
+ ClassiifyTab()
+ InitializeClassifyTab(Manager) : void
- InitializeAttributeList() : void
- InitializeValueComboBox() : void
- InitializeAttributeLabel() : void
- InitializeButtons() : void
- OnActionNextButton(object, EventArgs) : void
- OnActionBackButton(object, EventArgs) : void
- Classify(object, EventArgs) : void
- Reset(object, EventArgs) : void
- OnMouseEnter(object, EventArgs) : void
- OnMouseLeave(object, EventArgs) : void
- ChangeValueComboBox() : void
- MouseEnterColor<T : Button>(T) : void
- MouseLeavesColor<T : Button>(T) : void
<<UserControl>>
TableTab
+ TableTab()
+ InitializeTableTab(Manager) : void
- InitializeAttributeComboBox() : void
- GenerateTable() : void
- AttributeComboBoxSelectedIndexChanged(object, EventArgs) : void
- OnMouseHoverFilterbutton(object, EventArgs) : void
- OnMouseLeaveFilterButton(object, EventArgs) : void
- FilterButton_Click(object, EventArgs) : void
<<UserControl>>
ChartTab
+ ChartTab()
+ InitializeChartTab(Manager) : void
- GenerateTypeChart() : void
- GenerateOdorChart() : void
- GenerateRingNumberChart() : void
- GenerateBruisesChart() : void
- GenerateCapColorChart() : void
1
- form
<<use>>
<<MetroForm>>
MessageClassify
- attributes : List<string>
- values : List<char>
- classification : string
+ MesageClassify()
+ InitializeClassifyMessage(List<string>, List<char>, string, ClassifyTab) : void
- ShowMessage() : void
- OnFormClosed(object, FormClosedEventArgs) : void
Model
Manager
+ TRAINING_PERCENTAGE : double = 0.8
- decisionTreeOrg : DecisionTree
- codebook : Codification
- decisionTreeLib : DecisionTree
<<Property>> + DecisionTreeOrg : DecisionTree
<<Property>> + Codebook : Codification
<<Property>> + DecisionTreeLib : DecisionTree
+ Manager(string)
+ GenerateDecisionTreeOrg() : void
+ DecisionTreeAccuracyPercentageOrg() : double
+ GenerateDecisionTreeLib() : void
+ DecisionTreeAccuracyPercentageLib() : double
+ DecisionTreeClassifyLib(DataTable) : string[]
+ DecisionTreeDecisionsLib() : string
+ GenerateEmptyTable() : DataTable
+ GenerateDataTable() : DataTable
+ DataTableToMatrix(DataTable, string[]) : int[][]
+ GenerateFilteredDataTable(string, string) : DataTable
+ GenerateTrainingDataTableOrg() : DataTable
+ GenerateTestingDataTableOrg() : DataTable
+ GenerateEmptyTableLib() : DataTable
+ GenerateTrainingDataTableLib() : DataTable
+ GenerateTestingDataTableLib() : DataTable
+ GenerateTypeChart() : DataTable
+ GenerateOdorChart() : DataTable
+ GenerateRingNumberChart() : DataTable
+ GenerateBruisesChart() : DataTable
+ GenerateCapColorChart() : DataTable
+ Load(string) : int
- dataSet
0..*
Mushroom
+ CAP_SHAPE : char[] = {'b', 'c', 'x', 'f', 'k', 's'}
+CAP_SURFACE: char[] = {'f', 'g', 'y', 's'}
+CAP_COLOR: char[] = {'n', 'b', 'c', 'g', 'r', 'p', 'u', 'e', 'w', 'y' }
+BRUISES[]: char[] = {'t', 'f'}
+ODOR[]: char[] = {'a', 'l', 'c', 'y', 'f', 'm', 'n', 'p', 's'}
+GILL_ATTACHMENT: char[] = {'a', 'd', 'f', 'n'}
+GILL_SPACING: char[] = {'c', 'w', 'd'}
+GILL_SIZE: char[] = {'b', 'n'}
+GILL_COLOR: char[] = {'k', 'n', 'b', 'h', 'g', 'r', 'o', 'p', 'u', 'e', 'w', 'y'}
+STALK_SHAPE: char[] = {'e', 't'}
+STALK_ROOT: char[] = {'b', 'c', 'u', 'e', 'z', 'r', '?'}
+STALK_SURFACE_ABOVE_RING: char[] = {'f' , 'y', 'k', 's'}
+STALK_SURFACE_BELOW_RING: char[] = {'f' , 'y', 'k', 's'}
+STALK_COLOR_ABOVE_RING: char[] = {'n', 'b', 'c', 'g', 'o', 'p', 'e', 'w', 'y'}
+STALK_COLOR_BELOW_RING: char[] = {'n', 'b', 'c', 'g', 'o', 'p', 'e', 'w', 'y'}
+VEIL_TYPE: char[] = {´p', 'u'}
+VEIL_COLOR: char[] = {'n', 'o', 'w' , 'y'}
+RING_NUMBER: char[] = {'n', 'o', 't'}
+RING_TYPE: char[] = {'c', 'e', 'f', 'l', 'n', 'p', 's', 'z'}
+SPORE_PRINT_COLOR = {'k', 'n', 'b', 'h', 'r', 'o', 'u', 'w', 'y'}
+POPULATION: char[] = {'a', 'c', 'n', 's', 'v', 'y'}
+HABITAT: char[] = {'g', 'l', 'm', 'p', 'u', 'w', 'd'}
-type: MushroomType
-capShape:char
-capSurface:char
-capColor:char
-bruises:char
-odor:char
-gillAttachment:char
-gillSpacing:char
-gillSize: char
-gillColor: char
-stalkShape: char
-stalkRoot: char
-stalkSurfaceAboveRing: char
-stalkSurfaceBelowRing: char
-stalkColorAboveRing: char
-stalkColorBelowRing: char
-veilType: char
-veilColor: char
-ringNumber: char
-ringType: char
-sporePrintColor: char
-population: char
-habitat: char
<<Property>> +Type: MushroomType
<<Property>> +CapShape: char
<<Property>> +CapSurface: char
<<Property>> +CapColor: char
<<Property>> +Bruises: char
<<Property>> +Odor: char
<<Property>> +GillAttachment:char
<<Property>> +GillSpacing:char
<<Property>> +GillSize:char
<<Property>> +GillColor: char
<<Property>> +StalkShape: char
<<Property>> +StalkRoot: char
<<Property>> +StalkSurfaceAboveRing: char
<<Property>> +StalkSurfaceBelowRing: char
<<Property>> +StalkColorAboveRing: char
<<Property>> +StalkColorBelowRing: char
<<Property>> +VeilType: char
<<Property>> +VeilColor: char
<<Property>> +RingNumber: char
<<Property>> +RingType: char
<<Property>> +SporePrintColor: char
<<Property>> +Population: char
<<Property>> +Habitat: char
+ Mushroom(MushroomType,char,char,char,char,char,char,char,char,char,char,char,char,char,char,char,char,char,char,char,char,char,char)
+ VerifyMushroom(char,char,char,char,char,char,char,char,char,char,char,char,char,char,char,char,char,char,char,char,char,char) : bool
<<throw>>
Exception
<<SystemException>>
InvalidValuesException
+ InvalidValuesException()
+ InvalidValuesException(string)