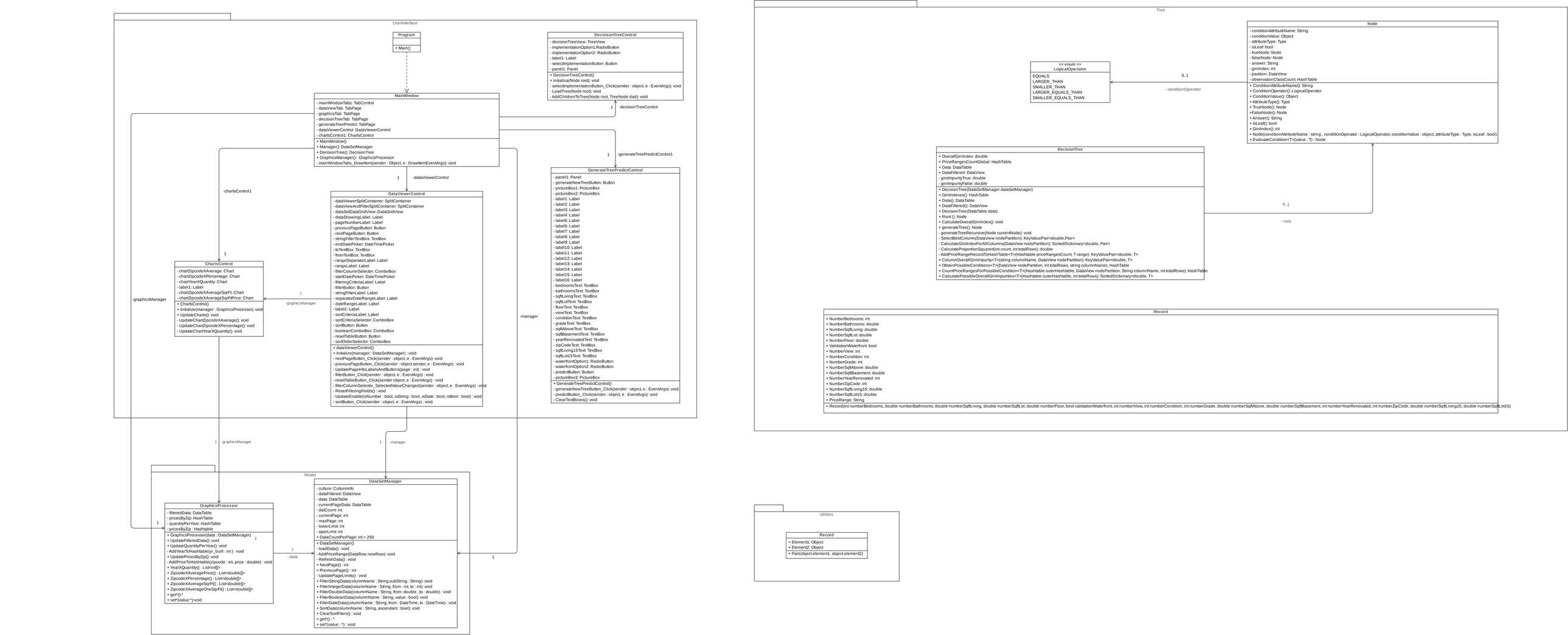
UserInterface
Program
+ Main()
DecisisonTreeControl
- decisionTreeView: TreeView
- implementationOption1:RadioButton
- implementationOption2: RadioButton
- label1: Label
- selectImplementationButton: Button
- panel1: Panel
+ DecisionTreeControl()
+ initializa(Node root): void
- selectImplementationButton_Click(sender : object, e : EventArgs): void
- LoadTree(Node root): void
- AddChildrenToTree(Node root, TreeNode dad): void
MainWindow
- mainWindowTabs: TabControl
- dataViewTab: TabPage
- graphicsTab: TabPage
- decisionTreeTab: TabPage
- generateTreePredict:TabPage
- dataViewerControl: DataViewerControl
- chartsControl1: ChartsControl
+ MainWindow()
+ Manager(): DataSetManager
+ DecisionTree(): DecisionTree
+ GraphicsManager(): GraphicsProcessor
- mainWindowTabs_DrawItem(sender : Object, e : DrawItemEventArgs): void
1
-decisionTreeControl
1
-generateTreePredictControl1
1
-dataViewerControl
-chartsControl1
1
-manager
GenerateTreePredictControl
- panel1: Panel
- generateNewTreeButton: Button
- pictureBox1: PictureBox
- pictureBox2: PictureBox
- label1: Label
- label2: Label
- label3: Label
- label4: Label
- label5: Label
- label6: Label
- label7: Label
- label8: Label
- label9: Label
- label10: Label
- label11: Label
- label12: Label
- label13: Label
- label14: Label
- label15: Label
- label16: Label
- bedroomsText: TextBox
- bathroomsText: TextBox
- sqftLivingText: TextBox
- sqftLotText: TextBox
- floorText: TextBox
- viewText: TextBox
- conditionText: TextBox
- gradeText: TextBox
- sqftAboveText: TextBox
- sqftBasementText: TextBox
- yearRenovatedText: TextBox
- zipCodeText: TextBox
- sqftLiving15Text: TextBox
- sqftLot15Text: TextBox
- waterfrontOption1: RadioButton
- waterfrontOption2: RadioButton
- predictButton: Button
- pictureBox3: PictureBox
+ GenerateTreePredictControl()
- generateNewTreeButton_Click(sender : object, e : EventArgs): void
- predictButton_Click(sender : object, e : EventArgs): void
- ClearTextBoxes(): void
DataViewerControl
- dataViewerSplitContainer: SplitContainer
- dataViewAndFilterSplitContainer: SplitContainer
- dataSetDataGridView: DataGridView
- dataShowingLabel: Label
- pageNumberLabel: Label
- prevousPageButton: Button
- nextPageButton: Button
- stringFilterTextBox: TextBox
- endDatePicker: DateTimePicker
- toTextBox: TextBox
- fromTextBox: TextBox
- rangeSeparatorLabel: Label
- rangeLabel: Label
- filterColumnSelector: ComboBox
- startDatePicker: DateTimePicker
- filteringCriteriaLabel: Label
- filterButton: Button
- stringFilterLabel: Label
- separatorDateRangeLabel: Label
- dateRangeLabel: Label
- label1: Label
- sortCriteriaLabel: Label
- sortCriteriaSelector: ComboBox
- sortButton: Button
- booleanComboBox: ComboBox
- resetTableButton: Button
- sortOrderSelector: ComboBox
+ dataViewerControl()
+ Initialize(manager : DataSetManager) : void
- nextPageButton_Click(sender : object, e : EventArgs): void
- prevousPageButton_Click(sender : object sender, e : EventArgs) : void
- UpdatePageInfoLabelsAndButtons(page : int) : void
- filterButton_Click(sender : object, e : EventArgs) : void
- resetTableButton_Click(sender object, e : EventArgs) : void
- filterColumnSelector_SelectedValueChanged(sender : object, e : EventArgs) : void
- ResetFilteringFields() : void
- UpdateEnable(isNumber : bool, isString : bool, isDate : bool, isBool : bool) : void
- sortButton_Click(sender : object, e : EventArgs) : void
ChartsControl
- chartZipcodeXAverage: Chart
- chartZipcodeXPercentage: Chart
- chartYearXQuantity: Chart
- label1: Label
- chartZipcodeXAverageSqrFt: Chart
- chartZipcodeXAverageSqrFtPrice: Chart
+ ChartsControl()
+ Initialize(manager : GraphicsProcessor): void
+ UpdateCharts(): void
- UpdateChartZipcodeXAverage(): void
- UpdateChartZipcodeXPercentage(): void
- UpdateChartYearXQuantity(): void
-graphicsManager
1
-graphicsManager
1
-graphicsManager
1
-manager
Model
DataSetManager
- culture: CultureInfo
- dataFiltered: DataView
- data: DataTable
- currentPageData: DataTable
- datCount: int
- currentPage: int
- maxPage: int
- lowerLimit: int
- operLimit: int
+ DataCountPerPage: int = 250
+ DataSetManager()
- loadData() : void
- AddPriceRange(DataRow newRow): void
- RefreshData() : void
+ NextPage() : int
+ PrevousPage() : int
- UpdatePageLimits() : void
+ FilterStringData(columnName : String,subString : String): void
+ FilterIntegerData(columnName : String, from : int, to : int): void
+ FilterDoubleData(columnName : String, from: double , to : double) : void
+ FilterBooleanData(columnName : String, value : bool): void
+ FilterDateData(columnName : String, from : DateTime, to : DateTime) : void
+ SortData(columnName : String, ascendant : bool): void
+ ClearSortFilters() : void
+ get*() .*
+ set*(value : *) : void
GraphicsProcessor
- filteredData: DataTable
- pricesByZip: HashTable
- quantityPerYear: HashTable
- pricesByZip : Hashtable
+ GraphicsProcessor(data : DataSetManager)
+ UpdateFilteredData(): void
+ UpdateQuantityPerYear(): void
- AddYearToHashable(yr_built : int ) : void
+ UpdatePricesByZip(): void
- AddPriceToHashable(zipcode : int, price : double) : void
+ YearXQuantity() : List<int[]>
+ ZipcodeXAveragePrice() : List<double[]>
+ ZipcodeXPercentage() : List<double[]>
+ ZipcodeXAverageSqrFt() : List<double[]>
+ ZipcodeXAverageOneSqrFt() : List<double[]>
+ get*():*
+ set*(value:*):void
1
1
- data
Tree
Node
- conditionAttributeName: String
- conditionValue: Object
- attributeType: Type
- isLeaf: bool
- trueNode: Node
- falseNode: Node
- answer: String
- giniIndex: int
- partition: DataView
- observationClassCount: HashTable
+ ConditionAttributeName(): String
+ ConditionOperator(): LogicalOperator
+ ConditionValue(): Object
+ AttributeType(): Type
+ TrueNode(): Node
+FalseNode(): Node
+ Answer(): String
+ isLeaf(): bool
+ GiniIndex(): int
+ Node(conditionAttributeName : string , conditionOperator : LogicalOperator, conditionValue : object, attributeType : Type, isLeaf : bool)
+ EvaluateCondition<T>(value : T) : Node
<< enum >>
LogicalOpertator
EQUALS
LARGER_THAN
SMALLER_THAN
LARGER_EQUALS_THAN
SMALLER_EQUALS_THAN
0..1
- conditionOperator
DecisionTree
+ OverallGiniIndex: double
+ PriceRangesCountGlobal: HashTable
+ Data: DataTable
+ DataFiltered: DataView
- giniImpurityTrue: double
- giniImpurityFalse: double
+ DecisionTree(DataSetManager dataSetManager)
+ GiniIndexes(): HashTable
+ Data(): DataTable
+ DataFiltered(): DataView
+ DecisionTree(DataTable data)
+ Root (): Node
+ CalculateOverallGiniIndex(): void
+ generateTree(): Node
- generateTreeRecursive(Node currentNode): void
- SelectBestColumn(DataView nodePartition): KeyValuePair<double,Pair>
- CalculateGiniIndexForAllColumns(DataView nodePartition): SortedDictionary<double, Pair>
- CalculateProportionSquared(int count, int totalRows): double
- AddPriceRangeRecordToHashTable<T>(Hashtable priceRangesCount, T range): KeyValuePair<double, T>
+ ColumnOverallGiniImpurity<T>(string columnName, DataView nodePartition): KeyValuePair<double, T>
+ ObtainPossibleConditions<T>(DataView nodePartition, int totalRows, string columnName): HashTable
+ CountPriceRangesForPossibleCondition<T>(Hashtable outerHashtable, DataView nodePartition, String columnName, int totalRows): HashTable
+ CalculatePossibleOverallGiniImpurities<T>(Hashtable outerHashtable, int totalRows): SortedDictionary<double, T>
0..1
- root
Record
+ NumberBedrooms: int
+ NumberBathrooms: double
+ NumberSqftLiving: double
+ NumberSqftLot: double
+ NumberFloor: double
+ ValidationWaterfront: bool
+ NumberView: int
+ NumberCondition: int
+ NumberGrade: int
+ NumberSqftAbove: double
+ NumberSqftBasement: double
+ NumberYearRenovated: int
+ NumberZipCode: int
+ NumberSqftLiving15: double
+ NumberSqftLot15: double
+ PriceRange: String
+ Record(int numberBedrooms, double numberBathrooms, double numberSqftLiving, double numberSqftLot, double numberFloor, bool validationWaterfront, int numberView, int numberCondition, int numberGrade, double numberSqftAbove, double numberSqftBasement, int numberYearRenovated, int numberZipCode, double numberSqftLiving15, double numberSqftLot15)
Utilities
Record
+ Element1: Object
+ Element2: Object
+ Pair(object element1, object element2)