#### Feature

stringValue : StringnumValue : FloatcorX : IntegercorY: IntegerrankList : RankfName : String

+ getStringValue(): String

+ setStringValue(stringValue : String) : void

+ getNumValue() : Float

+ setNumValue(numValue : Float) : void

+ getCorX(): Integer

+ setCorX(corX : Integer ) : void + setCorY(corY: Integer ) : void

+ getFName() : String

+ getRank() : int

+ toString(): String

## Rank

- name : String

- list : ArrayList<Feature>

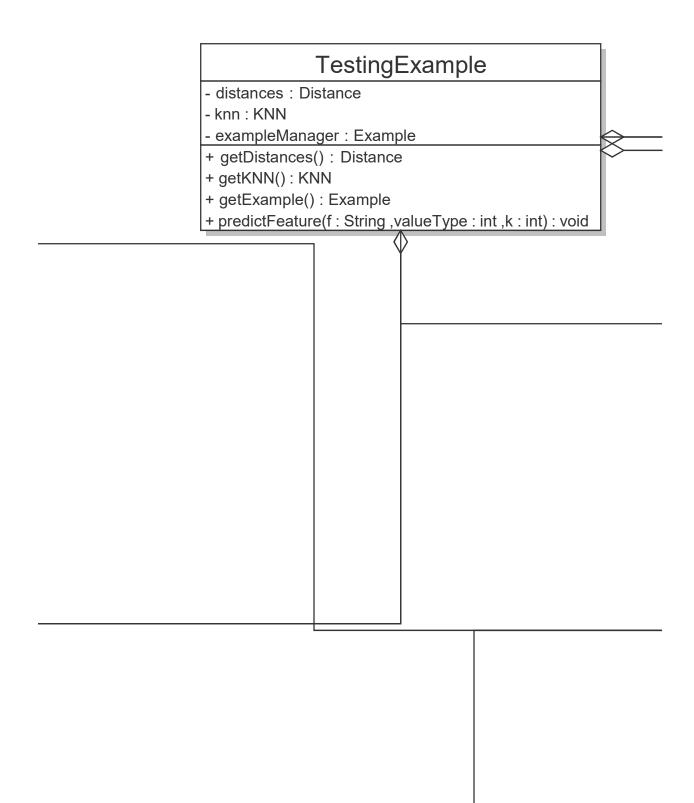
+ addInRank(feature : Feature) : void

+ getName(): String

+ checkName(name : String) : boolean

+ getlist() : ArrayList<Feature>
+ getValueAtRank(r: int) : String

+ getPosition(stringValue : String) : int



	⊤rainingExample	
	exampleName : String feature : HashMap <string,feature> nameSet : DefaultListModel<string></string></string,feature>	
	<ul> <li>createExample(name :String) : void</li> <li>addFeature(keyname : String,value : Feature) : voi</li> <li>getFeature(keyname : String) : Feature</li> <li>removeFeature(keyname : String) : void</li> <li>getExampleName() : String</li> </ul>	d
$\rightarrow$	<ul> <li>+ editFeature(keyname : String, value : Feature) : voi</li> <li>+editExampleName(name : String) : void</li> <li>+ checkKeyName(keyname: String ) : boolean</li> <li>+ getKeyName(value : Feature) : String</li> <li>+ getNameSet() : DefaultListModel<string></string></li> </ul>	d
	<ul> <li>getAllFeatures(): HashMap<string,feature></string,feature></li> <li>setFeatures(h: HashMap<string, feature="">): void</string,></li> <li>getTrainingExampleName(): String</li> <li>setTrainingExampleName(name: String): void</li> <li>toString(): String</li> </ul>	
L	r todanig() . Gaing	_

	- distances : Map <string,arrayl< th=""></string,arrayl<>	
	- normDist : Map <string,arrayli< th=""></string,arrayli<>	
	- lookUpTable : Map <traininge< th=""></traininge<>	
	- testEx : TestingExample	
	+ getDistance(train : Feature, te	
	+ findDistance(testExam : Testi	
	+ normalizeDistance() : void	
	+ updateDistances() : void	
	+ getEuclideanDistance( train :	
	+ getSubjectiveDistance( train :	
	+ getAbsoluteDistance( train : F	
	- nearestNeighbors : ArrayList <train< td=""></train<>	
	- k : int	
$\overline{}$	- testEx : TestingExample	
	+ getK(): int	
	+ setK(n : int ) : void	
	+ getNN() : ArrayList <trainingexamp< td=""></trainingexamp<>	
	+ determineNearestNeighbors(effK :	
	+ findKNN(effK : int,potentialNN : Arra	

# **KNNView**

- controller : KNNController

trainingPanel : JPaneltestingPanel : JPanel

- predict : JButton

- createTrainExample : JMe

- createTestExample : JMer

- editTrainFeature : JMenuIt

- addTrainingFeature : JMer

### Distance

ist<Float>> st<Float>> xample, Integer>

st : Feature) : Float

ngExample , trainExam : TrainingExample) : float

Feature, test : Feature) : float Feature, test : Feature) : float eature, test : Feature) : float

### **KNN**

ingExample>

le>

int,potentialNN : ArrayList<TrainingExample>) : void

ayList<TrainingExample>): void

nultem ıultem em

านltem

#### Example

- trainingExamples : DefaultListModel<TrainingE
- trainingExamplesModel : ArrayList<TrainingExa
- testingExamples : DefaultListModel<TestingExamples :
- type : ArrayList<String>
- rank : ArrayList<Rank>
- + addTrainingExample( example :TrainingExampl
- + getTrainingExamples() : DefaultListModel<Trai
- + getTrainingExamplesModel(): ArrayList<Trainii
- +addTestingExample( example : TestingExample
- + getTestingExample() : DefaultListModel<Testin
- + abstractkey(example: TrainingExample ) : void
- + createFeatureType(featureName : String , feature
- + appendRank( featureName: String, feature: Fea
- + getRankingList(fName : String ) : Rank
- + checkSubjective(feature: Feature): boolean
- + checkAbsolute( feature: Feature) : boolean
- + checkEuclidean( feature: Feature) : boolean
- + getTrainingExampleIndex(i : int) : TrainingExam
- + getTestingExampleIndex(i: int) : TestingExampl
- + toString(): String
- + calculateError( tEx : TrainingExample, f : Feature

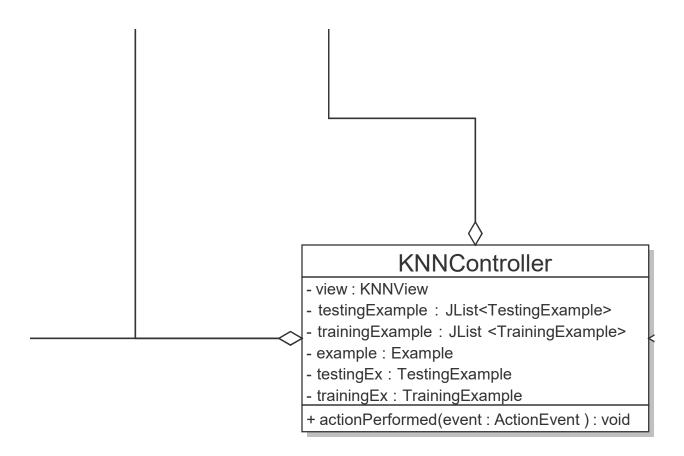
xample> mple> ample>

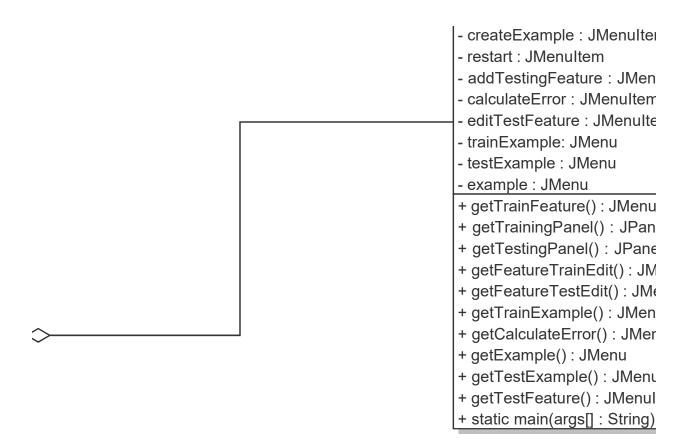
le): void IningExample> ngExample> ): void IgExample>

re : Feature) : void ıture) : void

ıple e

ક, k : int) : float





m

ultem

)

<del>;</del>m

Item

el

اڊ

lenuItem

enultem

ultem

านltem

ıltem

tem

: void