<<Interface>> extensibleclustering:: dependencies::Parser Interface that a parser will need to implement.

Responsibilities:

- + parseFile(file: File):
- Position[]
- + getName(): String
- + getDescription(): String

getSupportedFileExtensions (): ArrayList<String>



extensibleclustering:: dependencies::Position Stores data for a position. This includes the ID, and the components that make up the vector for the position.

Attributes

- ID: String
- Components: Double[]

Responsibilities

- <<constructor>> +
- Position(ID: String,
- components: Double[])
- + getID(): String
- + getComponents():
- Double[]
- + getDistance(otherPosition: Position):
- Double,
- + toString(): String
- + getDescription(): String
- + hashCode(): int
- + equals(obj: Object):
- boolean



<<exception>> extensibleclustering:: dependencies::

IncomparableComponentsException Exception that is thrown when the components of two positions are of different lengths and cannot be compared.

Methods:

<constructor>> + IncomparabelComponentsException

<<Interface>> extensibleclustering:: dependencies::ClusteringAlgorithm An interface that needs to be implemented by each clustering algorithm. The clusterData method should return the output file.

Methods:

- + clusterData(Position[] data): Centroid[]
- + getLastOutputFile(): File
- + getName(): String
- + getDescription(): String

extensibleclustering:: dependencies::Centroid Class for representing a single cluster

Attributes

- ID: String
- location: Position
- assignedPositions: Vector<Position>

Responsibilities

- <<constructor>> +
- Clusteroid(Id: String,
- location: Position)
- + getLocation(): Position
- + getID(): String
- + clearAssignedPositions():
- + assignPosition(position:
- Position): void
- + getAssignedPositions():
- Vector<Position>
- + centreCluster(): void
- + toString(): String
- + hashCode(): int
- + equals(obj: Object):

boolean

<<Interface>> extensibleclustering:: dependencies::Visualisation Interface that a visualisation method will need to implement.

Responsibilities:

- + visualiseData (Path: dataFile): Scene
- + getName(): String
- + getDescription(): String

extensibleclustering:: dependencies:: DirectoryHelper Helps check the required plugin, output etc directories exist and can be used. Can also get the path for them so the person creating the plugin does not have to do it.

- + <<constructor>> + DirectoryHelper()
- + createRequiredDirectories():
- + getPluginFolder(): Path
- + getOutputFolder(): Path
- correctFolderStructureExists(): boolean

extensibleclustering:: dependencies::Metrics Contains metric functions that can be used in the clustering process. These methods require centroids that have had data assigned to them

- + calinskiHarabasz (clusters: Centroid[], numPositions: Integer): Double
- + withinClusterVariance (clusters: Centroid[]): Double
- + betweenClusterVariance (clusters: Centroid[]): Double