密度聚类DBScan

**说明：密度聚类时根据点集之间的距离进行分组，密度聚类的实现方式有好几种，这里实现了DBScan，对密度聚类进行抽象和实现，可以不仅仅对点进行密度聚类：**

命名空间：ThMEPEngineCore.Algorithm.ClusteringAlgorithm

基础实现：DBScanClustering

说明：这里时将聚类的点抽象为图的节点，可以不只是对点进行聚类，可以将其它对象抽象成相应的数据，两个节点间的权重来代替两点间的距离，这里可以附近的聚类结果进行合并，合并后的个数也不会超过最多个数，超过的不会合并，优先将小的合并到附近的其它分组。

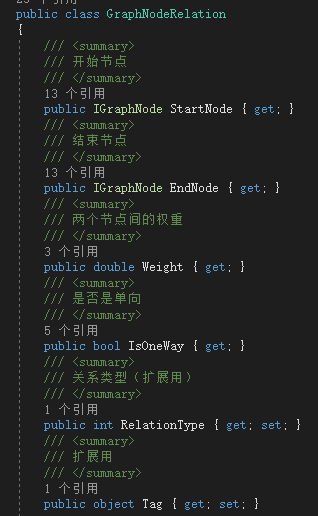
**优化点1：**可以让两个满足距离半径的点不聚在一起，这如果两个节点之间没有关系可以不传入两点间的关系，或传入两个点关系中的权重比较大，让聚类的时候不聚集在一起。

**优化点2：**可以限制每个聚类的最多个数，一个节点可以表示多个，通过设置节点的权重，目前一个节点只代表一个，权重多个的没有太多测试。

**优化点3：**不仅可以对点进行聚类，可以将其它对象抽象为节点进行密度聚类

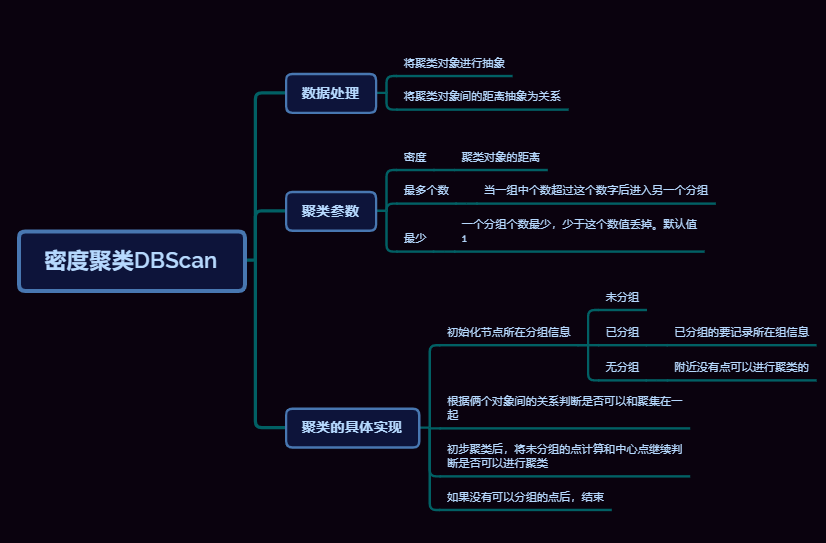
抽象节点:IGraphNode 该抽象节点可以用在部分图的算法中

节点关系：GraphNodeRelation 该抽象可以用在部分图的算法中，可以支持有向和无向图的关系表达



**如果是有向图，实在IsOneWay为true，注意起点和终点的的先后顺序**

将CAD中点集根据上述抽象的密度聚类进行聚类



1. 将CAD点抽象为聚类所需的节点

命名空间：ThMEPEngineCore.Algorithm.GraphDomain

抽象节点：PointGraphNode

1. 节点间的关系，抽象为图的节点关系

命名空间：ThMEPEngineCore.Algorithm.GraphDomain

抽象关系：GraphNodeRelation

该抽象关系可用于图的相关算法，可以支持有向图和无向图的关系，

1. CAD点进行密度聚类的实现

说明：这里根据点集+不可穿线进行分割，如果没有不可穿分割可以只根据点集

有不可穿线，如果需要聚类的两个点连线和该线有相交的，则认为这两个点不聚在一起

命名空间：ThMEPEngineCore.Algorithm.ClusteringAlgorithm

聚类实现：PointDBScan，内部再调用基础的聚类

先将点转为相应的抽象对象，再根据点和不可穿框线，构造相应的连接关系

