데이터사이언스

**DBSCAN implement assignment**

**2013011800** 구장회

**#** 순서

1. **data structure - DBSCAN,Pair,Point class**
2. **summary of algorithm**
3. **instruction for compiling**

**[**주의**]**

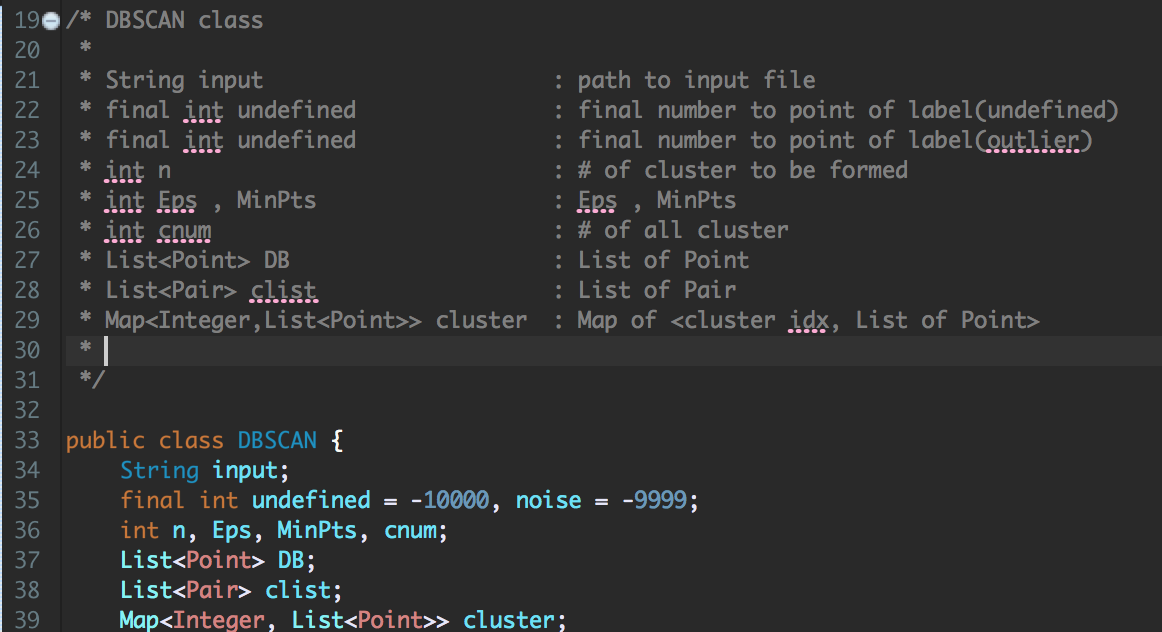
**[**모든스크린샷은첨부되어있음**]**

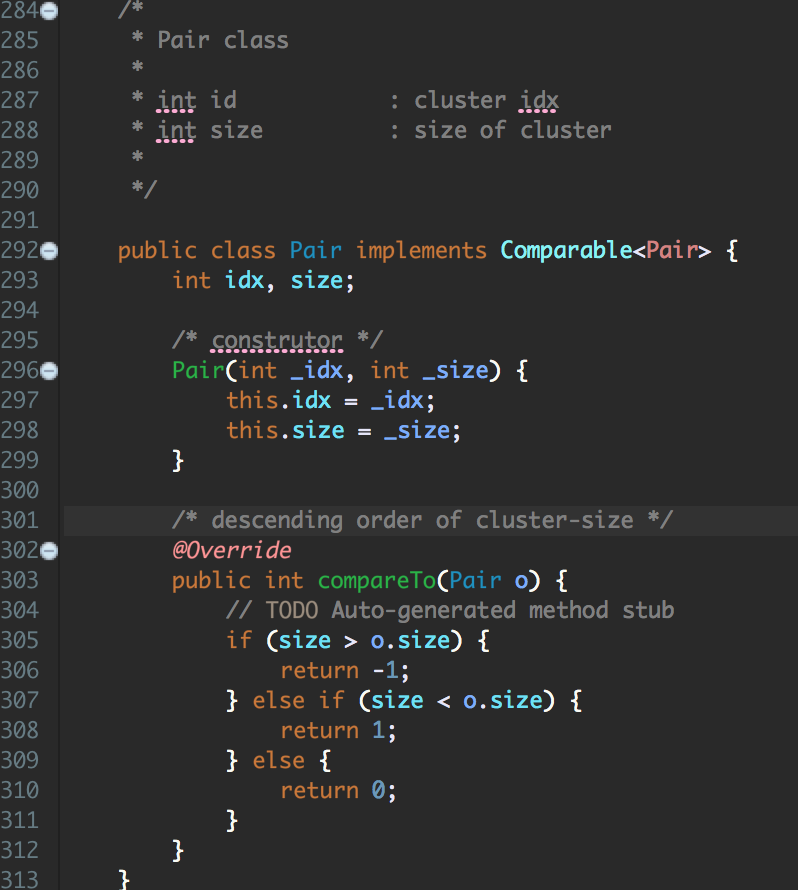
**[**실제로는 **dbscan package**에넣었지만**,** 제출시에는 **default package**로했음**]**

**[**실행파일은 **jar**파일로첨부했음**]**

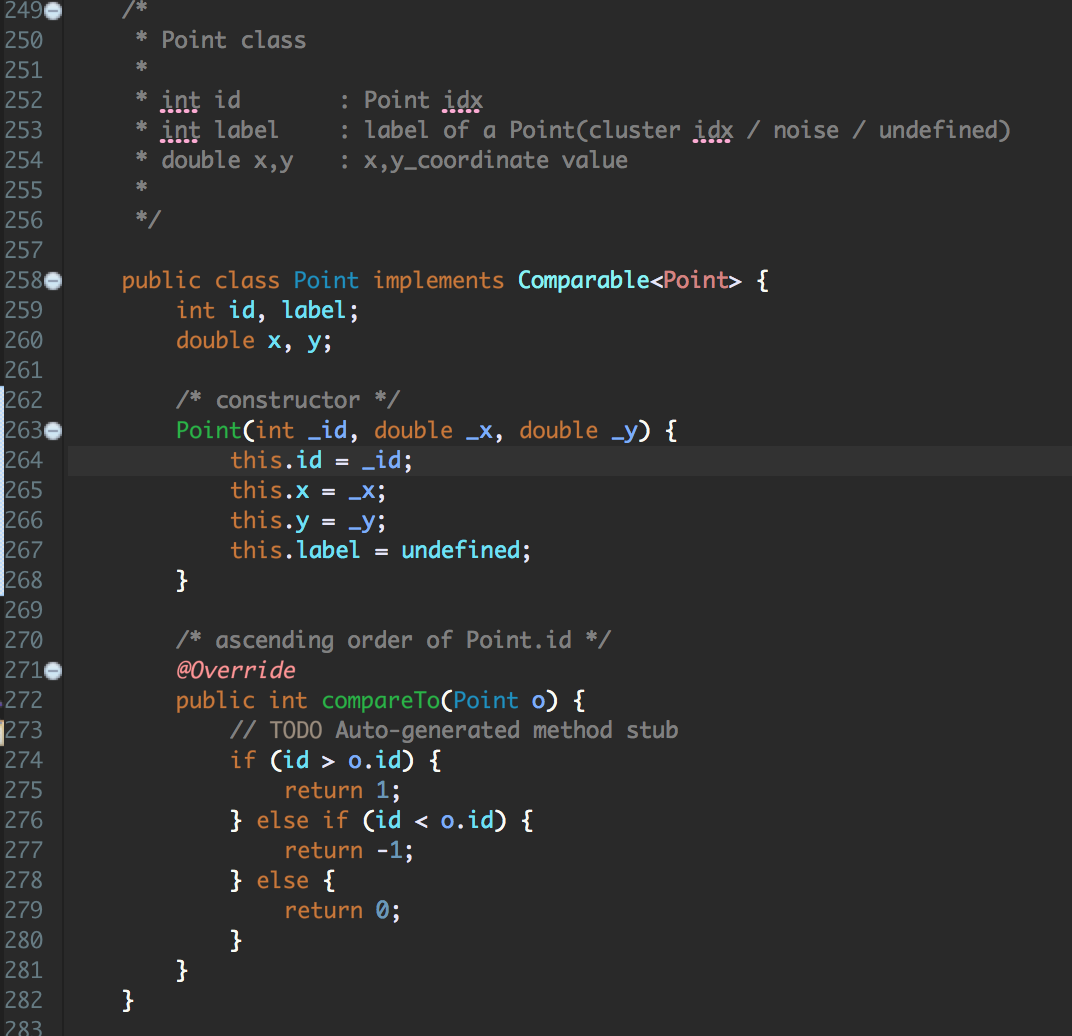
**# data structure**

**[DBSCAN class]**

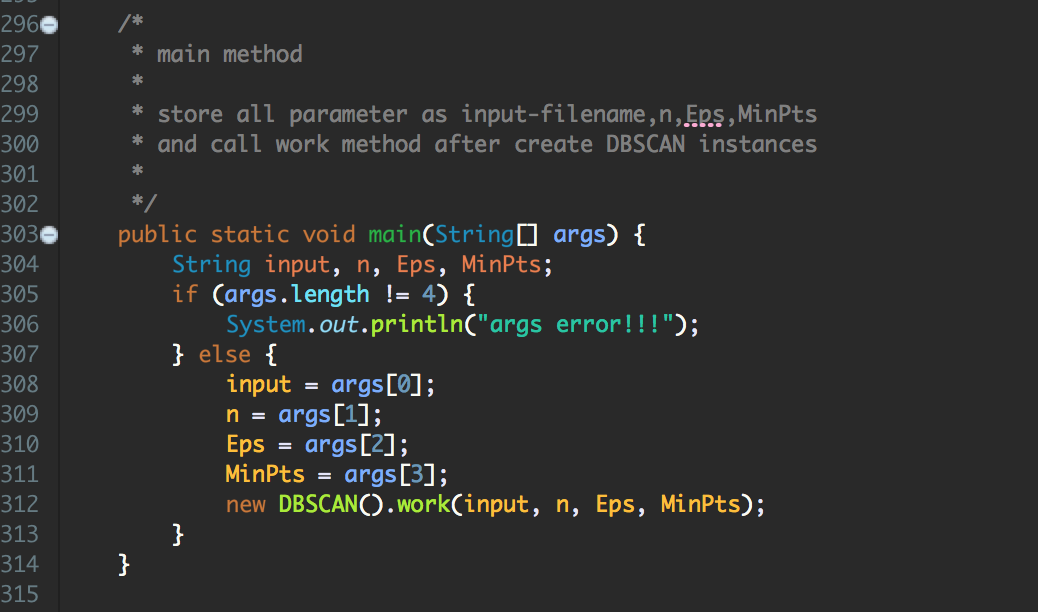
****

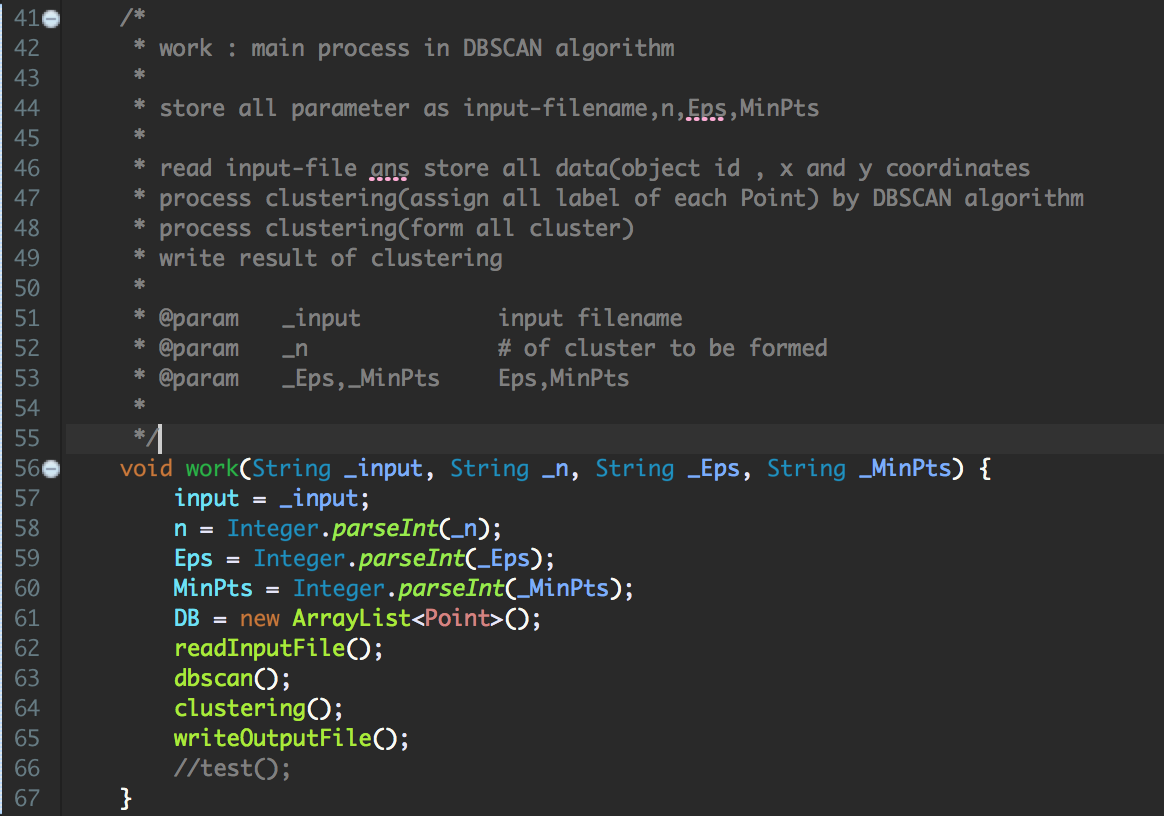
**[Pair class]**

**[Point class]**

****

**# summary of algorithms(flow)**

**[main method]**

**[main workflow]**

**# summary of algorithms(implemetation details)**

/\*

\* DBSCAN algorithm : assign all label of each Point

\*

\* for all Point which has no label,

\*

\* retrieve neighbor of the Point as set.

\* if neighbor's size < MinPts, then assign label as noise.

\* if not, the point being core-point.

\* so increment cluster idx(cnum).

\* and enqueue all neighbor except core-point.

\*

\* for all Point in queue,

\*

\* if the Point's label is noise,

\* then assign label as same cluster idx before.

\* if the Point's label is undefined,

\* then continue.

\* enqueue all neighbor of the Point which are dense.

\*

\*/

void dbscan() {}(

/\*

\* clustering : form all cluster

\*

\* make clist as list<(cluster-idx,size of cluster)>.

\* make cmap as map<cluster-idx,list of Points>.

\*

\*/

void clustering() {}() {

/\*

\* write clustering result in output file

\*/

void writeOutputFile {}

/\*

\* RangeQuery : find neighbor

\*

\* for all Point q,

\* if dist(p,q) <= Eps, add to set of Point(neighbor)

\*

\* return set of neighbors

\*

\*

\* @param p Point as a center.

\* @return Set<Point> set of neighbors

\*/

Set<Point> RangeQuery(Point p) {}

{ /\*

\* read inputfile

\*/

void readInputFile {}) {

/\*

\* distance between two Points as Euclidean distance in 2d.

\*

\*/

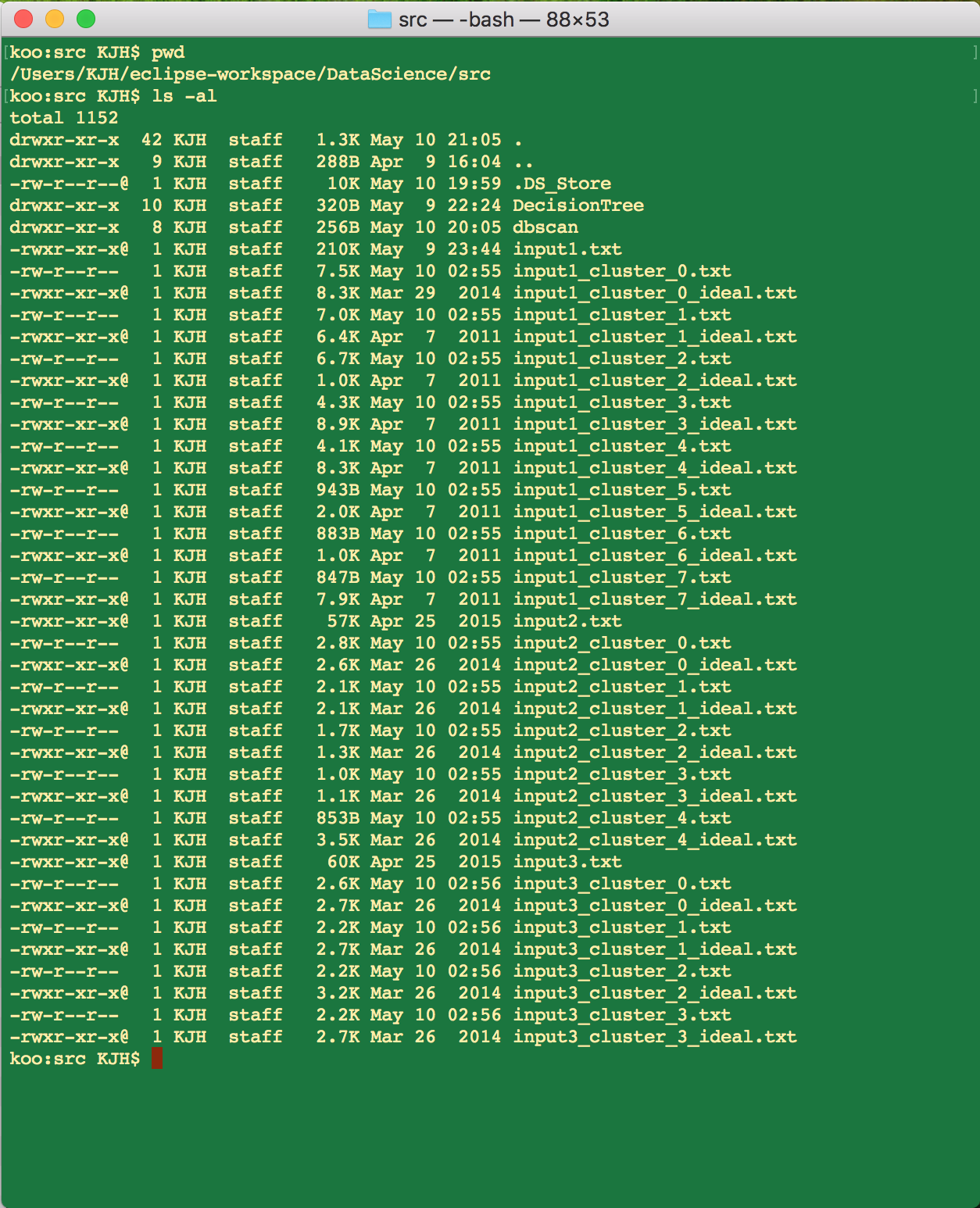
double dist(Point p, Point q) {}

**# instruction for compiling**

**[Envoirments]**

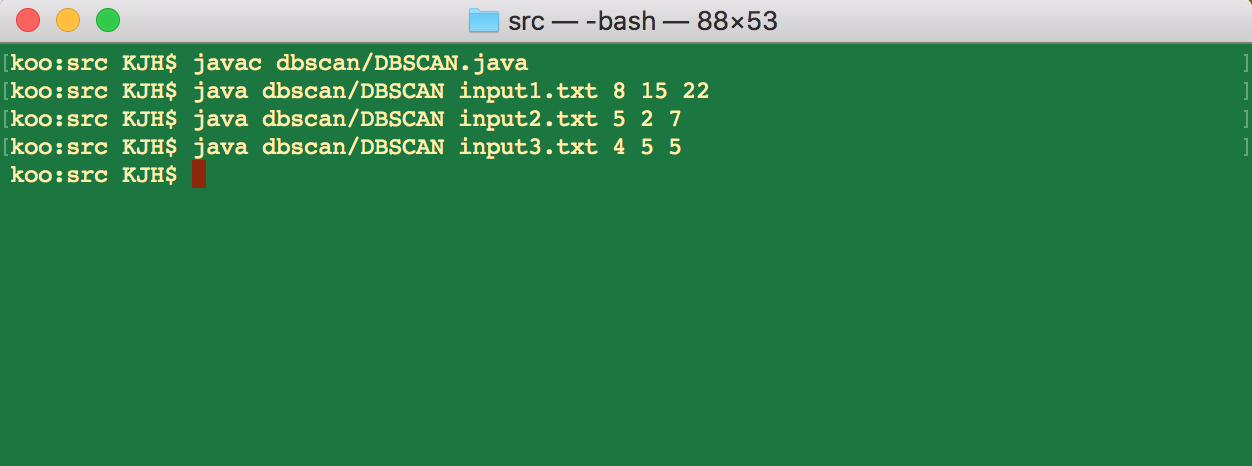
**OS : Mac OS**

**Language : java**

**[Screenshot-**실행전**]**

**[screenshot-**컴파일**,**실행**]**

컴파일 **: $ javac dbscan/DBSCAN.java**

실행 **: $ java dbscan/DBSCAN [inputfile] [n] [Eps] [MinPts]**