﻿import scala.io.StdIn.readLine;

class Kmeans {

var nombreItter: Int = 100;

var nombreClusters : Int = 0;

var points : Array[Array[Int]] = Array();

// Fonction qui crée des clusters aleatoirement

// crée @nombreClusters clusters

def creerClusters() : Array[Array[Int]]= {

var cluster : Array[Array[Int]] = Array.ofDim(this.nombreClusters);

val rnd = new scala.util.Random ;

var i : Int = 0;

while(i < this.nombreClusters)

{

var x = 1 + rnd.nextInt(10) ;

var y = 1 + rnd.nextInt(10);

var clust : Array[Int] = Array(x,y);

cluster(i) = clust;

i+=1;

}

println("avant MAJ");

for(i <- 0 to cluster.length-1)

{

for(j <- 0 to cluster(i).length-1)

{

print(cluster(i)(j)+" ");

}

println(" ");

}

println(" ");

return cluster;

}

// Fonction qui calcul la distance la plus proche entre les clusters et un point et renvoie

// le numero de cluster le plus proche du point

def calculerDistance(clusters : Array[Array[Int]], point : Array[Int]) : Int = {

var tmp : Double = Math.sqrt(Math.pow(clusters(0)(0)-point(0), 2)+Math.pow(clusters(0)(1)-point(1), 2));

var numeroCluster : Int = 0;

for ( i <- 1 to clusters.length-1)

{

if (Math.sqrt(Math.pow(clusters(i)(0)-point(0), 2)+Math.pow(clusters(i)(1)-point(1), 2)) < tmp )

{

tmp = Math.sqrt(Math.pow(clusters(i)(0)-point(0), 2)+Math.pow(clusters(i)(1)-point(1), 2));

numeroCluster = i;

}

}

//println(numeroCluster);

return numeroCluster;

}

// met a jour les coordonnées des clusters grace a la moyenne des points

def miseAJourCentre(clusters : Array[Array[Int]])= {

var c : Array[Int] = Array.ofDim(clusters.length);

for (i <- 0 to clusters.length-1)

{

var tmp : Int = 1;

for(j <- 0 to this.points.length-1)

{

if (calculerDistance(clusters, this.points(j)) == i)

{

//println(calculerDistance(clusters : Array[Array[Int]], this.points(j)));

clusters(i)(0) += this.points(j)(0) ;

clusters(i)(1) += this.points(j)(1);

tmp += 1;

}

}

clusters(i)(0) /= tmp;

clusters(i)(1) /= tmp;

}

println("apres MAJ");

for(i <- 0 to clusters.length-1)

{

for(j <- 0 to clusters(i).length-1)

{

print(clusters(i)(j)+" ");

}

println(" ");

}

}

//---------- Setters ----------//

def setNombreClusters(nbClusters : Int) = {

this.nombreClusters = nbClusters;

}

def setNombreItter(nbItter : Int) = {

this.nombreItter = nbItter;

}

def setPoints(points : Array[Array[Int]]) = {

this.points = points;

}

}