/\* ----------------------------------------------------------------------------

\* This file was automatically generated by SWIG (http://www.swig.org).

\* Version 4.0.2

\*

\* Do not make changes to this file unless you know what you are doing--modify

\* the SWIG interface file instead.

\* ----------------------------------------------------------------------------- \*/

package com.twitter.ann.faiss;

public class Clustering extends ClusteringParameters {

private transient long swigCPtr;

protected Clustering(long cPtr, boolean cMemoryOwn) {

super(swigfaissJNI.Clustering\_SWIGUpcast(cPtr), cMemoryOwn);

swigCPtr = cPtr;

}

protected static long getCPtr(Clustering obj) {

return (obj == null) ? 0 : obj.swigCPtr;

}

@SuppressWarnings("deprecation")

protected void finalize() {

delete();

}

public synchronized void delete() {

if (swigCPtr != 0) {

if (swigCMemOwn) {

swigCMemOwn = false;

swigfaissJNI.delete\_Clustering(swigCPtr);

}

swigCPtr = 0;

}

super.delete();

}

public void setD(long value) {

swigfaissJNI.Clustering\_d\_set(swigCPtr, this, value);

}

public long getD() {

return swigfaissJNI.Clustering\_d\_get(swigCPtr, this);

}

public void setK(long value) {

swigfaissJNI.Clustering\_k\_set(swigCPtr, this, value);

}

public long getK() {

return swigfaissJNI.Clustering\_k\_get(swigCPtr, this);

}

public void setCentroids(FloatVector value) {

swigfaissJNI.Clustering\_centroids\_set(swigCPtr, this, FloatVector.getCPtr(value), value);

}

public FloatVector getCentroids() {

long cPtr = swigfaissJNI.Clustering\_centroids\_get(swigCPtr, this);

return (cPtr == 0) ? null : new FloatVector(cPtr, false);

}

public void setIteration\_stats(SWIGTYPE\_p\_std\_\_vectorT\_faiss\_\_ClusteringIterationStats\_t value) {

swigfaissJNI.Clustering\_iteration\_stats\_set(swigCPtr, this, SWIGTYPE\_p\_std\_\_vectorT\_faiss\_\_ClusteringIterationStats\_t.getCPtr(value));

}

public SWIGTYPE\_p\_std\_\_vectorT\_faiss\_\_ClusteringIterationStats\_t getIteration\_stats() {

long cPtr = swigfaissJNI.Clustering\_iteration\_stats\_get(swigCPtr, this);

return (cPtr == 0) ? null : new SWIGTYPE\_p\_std\_\_vectorT\_faiss\_\_ClusteringIterationStats\_t(cPtr, false);

}

public Clustering(int d, int k) {

this(swigfaissJNI.new\_Clustering\_\_SWIG\_0(d, k), true);

}

public Clustering(int d, int k, ClusteringParameters cp) {

this(swigfaissJNI.new\_Clustering\_\_SWIG\_1(d, k, ClusteringParameters.getCPtr(cp), cp), true);

}

public void train(long n, SWIGTYPE\_p\_float x, Index index, SWIGTYPE\_p\_float x\_weights) {

swigfaissJNI.Clustering\_train\_\_SWIG\_0(swigCPtr, this, n, SWIGTYPE\_p\_float.getCPtr(x), Index.getCPtr(index), index, SWIGTYPE\_p\_float.getCPtr(x\_weights));

}

public void train(long n, SWIGTYPE\_p\_float x, Index index) {

swigfaissJNI.Clustering\_train\_\_SWIG\_1(swigCPtr, this, n, SWIGTYPE\_p\_float.getCPtr(x), Index.getCPtr(index), index);

}

public void train\_encoded(long nx, SWIGTYPE\_p\_unsigned\_char x\_in, Index codec, Index index, SWIGTYPE\_p\_float weights) {

swigfaissJNI.Clustering\_train\_encoded\_\_SWIG\_0(swigCPtr, this, nx, SWIGTYPE\_p\_unsigned\_char.getCPtr(x\_in), Index.getCPtr(codec), codec, Index.getCPtr(index), index, SWIGTYPE\_p\_float.getCPtr(weights));

}

public void train\_encoded(long nx, SWIGTYPE\_p\_unsigned\_char x\_in, Index codec, Index index) {

swigfaissJNI.Clustering\_train\_encoded\_\_SWIG\_1(swigCPtr, this, nx, SWIGTYPE\_p\_unsigned\_char.getCPtr(x\_in), Index.getCPtr(codec), codec, Index.getCPtr(index), index);

}

public void post\_process\_centroids() {

swigfaissJNI.Clustering\_post\_process\_centroids(swigCPtr, this);

}

}