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

\* 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 MultiIndexQuantizer extends Index {

private transient long swigCPtr;

protected MultiIndexQuantizer(long cPtr, boolean cMemoryOwn) {

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

swigCPtr = cPtr;

}

protected static long getCPtr(MultiIndexQuantizer 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\_MultiIndexQuantizer(swigCPtr);

}

swigCPtr = 0;

}

super.delete();

}

public void setPq(ProductQuantizer value) {

swigfaissJNI.MultiIndexQuantizer\_pq\_set(swigCPtr, this, ProductQuantizer.getCPtr(value), value);

}

public ProductQuantizer getPq() {

long cPtr = swigfaissJNI.MultiIndexQuantizer\_pq\_get(swigCPtr, this);

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

}

public MultiIndexQuantizer(int d, long M, long nbits) {

this(swigfaissJNI.new\_MultiIndexQuantizer\_\_SWIG\_0(d, M, nbits), true);

}

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

swigfaissJNI.MultiIndexQuantizer\_train(swigCPtr, this, n, SWIGTYPE\_p\_float.getCPtr(x));

}

public void search(long n, SWIGTYPE\_p\_float x, long k, SWIGTYPE\_p\_float distances, LongVector labels) {

swigfaissJNI.MultiIndexQuantizer\_search(swigCPtr, this, n, SWIGTYPE\_p\_float.getCPtr(x), k, SWIGTYPE\_p\_float.getCPtr(distances), SWIGTYPE\_p\_long\_long.getCPtr(labels.data()), labels);

}

public void add(long n, SWIGTYPE\_p\_float x) {

swigfaissJNI.MultiIndexQuantizer\_add(swigCPtr, this, n, SWIGTYPE\_p\_float.getCPtr(x));

}

public void reset() {

swigfaissJNI.MultiIndexQuantizer\_reset(swigCPtr, this);

}

public MultiIndexQuantizer() {

this(swigfaissJNI.new\_MultiIndexQuantizer\_\_SWIG\_1(), true);

}

public void reconstruct(long key, SWIGTYPE\_p\_float recons) {

swigfaissJNI.MultiIndexQuantizer\_reconstruct(swigCPtr, this, key, SWIGTYPE\_p\_float.getCPtr(recons));

}

}