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

\* 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 IndexShards {

private transient long swigCPtr;

protected transient boolean swigCMemOwn;

protected IndexShards(long cPtr, boolean cMemoryOwn) {

swigCMemOwn = cMemoryOwn;

swigCPtr = cPtr;

}

protected static long getCPtr(IndexShards 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\_IndexShards(swigCPtr);

}

swigCPtr = 0;

}

}

public IndexShards(boolean threaded, boolean successive\_ids) {

this(swigfaissJNI.new\_IndexShards\_\_SWIG\_0(threaded, successive\_ids), true);

}

public IndexShards(boolean threaded) {

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

}

public IndexShards() {

this(swigfaissJNI.new\_IndexShards\_\_SWIG\_2(), true);

}

public IndexShards(int d, boolean threaded, boolean successive\_ids) {

this(swigfaissJNI.new\_IndexShards\_\_SWIG\_3(d, threaded, successive\_ids), true);

}

public IndexShards(int d, boolean threaded) {

this(swigfaissJNI.new\_IndexShards\_\_SWIG\_4(d, threaded), true);

}

public IndexShards(int d) {

this(swigfaissJNI.new\_IndexShards\_\_SWIG\_5(d), true);

}

public void add\_shard(Index index) {

swigfaissJNI.IndexShards\_add\_shard(swigCPtr, this, Index.getCPtr(index), index);

}

public void remove\_shard(Index index) {

swigfaissJNI.IndexShards\_remove\_shard(swigCPtr, this, Index.getCPtr(index), index);

}

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

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

}

public void add\_with\_ids(long n, SWIGTYPE\_p\_float x, LongVector xids) {

swigfaissJNI.IndexShards\_add\_with\_ids(swigCPtr, this, n, SWIGTYPE\_p\_float.getCPtr(x), SWIGTYPE\_p\_long\_long.getCPtr(xids.data()), xids);

}

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

swigfaissJNI.IndexShards\_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 train(long n, SWIGTYPE\_p\_float x) {

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

}

public void setSuccessive\_ids(boolean value) {

swigfaissJNI.IndexShards\_successive\_ids\_set(swigCPtr, this, value);

}

public boolean getSuccessive\_ids() {

return swigfaissJNI.IndexShards\_successive\_ids\_get(swigCPtr, this);

}

public void syncWithSubIndexes() {

swigfaissJNI.IndexShards\_syncWithSubIndexes(swigCPtr, this);

}

}