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

\* 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 IndexSplitVectors extends Index {

private transient long swigCPtr;

protected IndexSplitVectors(long cPtr, boolean cMemoryOwn) {

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

swigCPtr = cPtr;

}

protected static long getCPtr(IndexSplitVectors 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\_IndexSplitVectors(swigCPtr);

}

swigCPtr = 0;

}

super.delete();

}

public void setOwn\_fields(boolean value) {

swigfaissJNI.IndexSplitVectors\_own\_fields\_set(swigCPtr, this, value);

}

public boolean getOwn\_fields() {

return swigfaissJNI.IndexSplitVectors\_own\_fields\_get(swigCPtr, this);

}

public void setThreaded(boolean value) {

swigfaissJNI.IndexSplitVectors\_threaded\_set(swigCPtr, this, value);

}

public boolean getThreaded() {

return swigfaissJNI.IndexSplitVectors\_threaded\_get(swigCPtr, this);

}

public void setSub\_indexes(SWIGTYPE\_p\_std\_\_vectorT\_faiss\_\_Index\_p\_t value) {

swigfaissJNI.IndexSplitVectors\_sub\_indexes\_set(swigCPtr, this, SWIGTYPE\_p\_std\_\_vectorT\_faiss\_\_Index\_p\_t.getCPtr(value));

}

public SWIGTYPE\_p\_std\_\_vectorT\_faiss\_\_Index\_p\_t getSub\_indexes() {

long cPtr = swigfaissJNI.IndexSplitVectors\_sub\_indexes\_get(swigCPtr, this);

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

}

public void setSum\_d(long value) {

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

}

public long getSum\_d() {

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

}

public void add\_sub\_index(Index arg0) {

swigfaissJNI.IndexSplitVectors\_add\_sub\_index(swigCPtr, this, Index.getCPtr(arg0), arg0);

}

public void sync\_with\_sub\_indexes() {

swigfaissJNI.IndexSplitVectors\_sync\_with\_sub\_indexes(swigCPtr, this);

}

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

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

}

public void reset() {

swigfaissJNI.IndexSplitVectors\_reset(swigCPtr, this);

}

}