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

\* 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 IndexBinaryFlat extends IndexBinary {

private transient long swigCPtr;

protected IndexBinaryFlat(long cPtr, boolean cMemoryOwn) {

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

swigCPtr = cPtr;

}

protected static long getCPtr(IndexBinaryFlat 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\_IndexBinaryFlat(swigCPtr);

}

swigCPtr = 0;

}

super.delete();

}

public void setXb(ByteVector value) {

swigfaissJNI.IndexBinaryFlat\_xb\_set(swigCPtr, this, ByteVector.getCPtr(value), value);

}

public ByteVector getXb() {

long cPtr = swigfaissJNI.IndexBinaryFlat\_xb\_get(swigCPtr, this);

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

}

public void setUse\_heap(boolean value) {

swigfaissJNI.IndexBinaryFlat\_use\_heap\_set(swigCPtr, this, value);

}

public boolean getUse\_heap() {

return swigfaissJNI.IndexBinaryFlat\_use\_heap\_get(swigCPtr, this);

}

public void setQuery\_batch\_size(long value) {

swigfaissJNI.IndexBinaryFlat\_query\_batch\_size\_set(swigCPtr, this, value);

}

public long getQuery\_batch\_size() {

return swigfaissJNI.IndexBinaryFlat\_query\_batch\_size\_get(swigCPtr, this);

}

public IndexBinaryFlat(long d) {

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

}

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

swigfaissJNI.IndexBinaryFlat\_add(swigCPtr, this, n, SWIGTYPE\_p\_unsigned\_char.getCPtr(x));

}

public void reset() {

swigfaissJNI.IndexBinaryFlat\_reset(swigCPtr, this);

}

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

swigfaissJNI.IndexBinaryFlat\_search(swigCPtr, this, n, SWIGTYPE\_p\_unsigned\_char.getCPtr(x), k, SWIGTYPE\_p\_int.getCPtr(distances), SWIGTYPE\_p\_long\_long.getCPtr(labels.data()), labels);

}

public void range\_search(long n, SWIGTYPE\_p\_unsigned\_char x, int radius, RangeSearchResult result) {

swigfaissJNI.IndexBinaryFlat\_range\_search(swigCPtr, this, n, SWIGTYPE\_p\_unsigned\_char.getCPtr(x), radius, RangeSearchResult.getCPtr(result), result);

}

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

swigfaissJNI.IndexBinaryFlat\_reconstruct(swigCPtr, this, key, SWIGTYPE\_p\_unsigned\_char.getCPtr(recons));

}

public long remove\_ids(IDSelector sel) {

return swigfaissJNI.IndexBinaryFlat\_remove\_ids(swigCPtr, this, IDSelector.getCPtr(sel), sel);

}

public IndexBinaryFlat() {

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

}

}