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

\* 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 LinearTransform extends VectorTransform {

private transient long swigCPtr;

protected LinearTransform(long cPtr, boolean cMemoryOwn) {

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

swigCPtr = cPtr;

}

protected static long getCPtr(LinearTransform 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\_LinearTransform(swigCPtr);

}

swigCPtr = 0;

}

super.delete();

}

public void setHave\_bias(boolean value) {

swigfaissJNI.LinearTransform\_have\_bias\_set(swigCPtr, this, value);

}

public boolean getHave\_bias() {

return swigfaissJNI.LinearTransform\_have\_bias\_get(swigCPtr, this);

}

public void setIs\_orthonormal(boolean value) {

swigfaissJNI.LinearTransform\_is\_orthonormal\_set(swigCPtr, this, value);

}

public boolean getIs\_orthonormal() {

return swigfaissJNI.LinearTransform\_is\_orthonormal\_get(swigCPtr, this);

}

public void setA(FloatVector value) {

swigfaissJNI.LinearTransform\_A\_set(swigCPtr, this, FloatVector.getCPtr(value), value);

}

public FloatVector getA() {

long cPtr = swigfaissJNI.LinearTransform\_A\_get(swigCPtr, this);

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

}

public void setB(FloatVector value) {

swigfaissJNI.LinearTransform\_b\_set(swigCPtr, this, FloatVector.getCPtr(value), value);

}

public FloatVector getB() {

long cPtr = swigfaissJNI.LinearTransform\_b\_get(swigCPtr, this);

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

}

public LinearTransform(int d\_in, int d\_out, boolean have\_bias) {

this(swigfaissJNI.new\_LinearTransform\_\_SWIG\_0(d\_in, d\_out, have\_bias), true);

}

public LinearTransform(int d\_in, int d\_out) {

this(swigfaissJNI.new\_LinearTransform\_\_SWIG\_1(d\_in, d\_out), true);

}

public LinearTransform(int d\_in) {

this(swigfaissJNI.new\_LinearTransform\_\_SWIG\_2(d\_in), true);

}

public LinearTransform() {

this(swigfaissJNI.new\_LinearTransform\_\_SWIG\_3(), true);

}

public void apply\_noalloc(long n, SWIGTYPE\_p\_float x, SWIGTYPE\_p\_float xt) {

swigfaissJNI.LinearTransform\_apply\_noalloc(swigCPtr, this, n, SWIGTYPE\_p\_float.getCPtr(x), SWIGTYPE\_p\_float.getCPtr(xt));

}

public void transform\_transpose(long n, SWIGTYPE\_p\_float y, SWIGTYPE\_p\_float x) {

swigfaissJNI.LinearTransform\_transform\_transpose(swigCPtr, this, n, SWIGTYPE\_p\_float.getCPtr(y), SWIGTYPE\_p\_float.getCPtr(x));

}

public void reverse\_transform(long n, SWIGTYPE\_p\_float xt, SWIGTYPE\_p\_float x) {

swigfaissJNI.LinearTransform\_reverse\_transform(swigCPtr, this, n, SWIGTYPE\_p\_float.getCPtr(xt), SWIGTYPE\_p\_float.getCPtr(x));

}

public void set\_is\_orthonormal() {

swigfaissJNI.LinearTransform\_set\_is\_orthonormal(swigCPtr, this);

}

public void setVerbose(boolean value) {

swigfaissJNI.LinearTransform\_verbose\_set(swigCPtr, this, value);

}

public boolean getVerbose() {

return swigfaissJNI.LinearTransform\_verbose\_get(swigCPtr, this);

}

public void print\_if\_verbose(String name, DoubleVector mat, int n, int d) {

swigfaissJNI.LinearTransform\_print\_if\_verbose(swigCPtr, this, name, DoubleVector.getCPtr(mat), mat, n, d);

}

}