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

\* 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 ParameterSpace {

private transient long swigCPtr;

protected transient boolean swigCMemOwn;

protected ParameterSpace(long cPtr, boolean cMemoryOwn) {

swigCMemOwn = cMemoryOwn;

swigCPtr = cPtr;

}

protected static long getCPtr(ParameterSpace 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\_ParameterSpace(swigCPtr);

}

swigCPtr = 0;

}

}

public void setParameter\_ranges(SWIGTYPE\_p\_std\_\_vectorT\_faiss\_\_ParameterRange\_t value) {

swigfaissJNI.ParameterSpace\_parameter\_ranges\_set(swigCPtr, this, SWIGTYPE\_p\_std\_\_vectorT\_faiss\_\_ParameterRange\_t.getCPtr(value));

}

public SWIGTYPE\_p\_std\_\_vectorT\_faiss\_\_ParameterRange\_t getParameter\_ranges() {

long cPtr = swigfaissJNI.ParameterSpace\_parameter\_ranges\_get(swigCPtr, this);

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

}

public void setVerbose(int value) {

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

}

public int getVerbose() {

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

}

public void setN\_experiments(int value) {

swigfaissJNI.ParameterSpace\_n\_experiments\_set(swigCPtr, this, value);

}

public int getN\_experiments() {

return swigfaissJNI.ParameterSpace\_n\_experiments\_get(swigCPtr, this);

}

public void setBatchsize(long value) {

swigfaissJNI.ParameterSpace\_batchsize\_set(swigCPtr, this, value);

}

public long getBatchsize() {

return swigfaissJNI.ParameterSpace\_batchsize\_get(swigCPtr, this);

}

public void setThread\_over\_batches(boolean value) {

swigfaissJNI.ParameterSpace\_thread\_over\_batches\_set(swigCPtr, this, value);

}

public boolean getThread\_over\_batches() {

return swigfaissJNI.ParameterSpace\_thread\_over\_batches\_get(swigCPtr, this);

}

public void setMin\_test\_duration(double value) {

swigfaissJNI.ParameterSpace\_min\_test\_duration\_set(swigCPtr, this, value);

}

public double getMin\_test\_duration() {

return swigfaissJNI.ParameterSpace\_min\_test\_duration\_get(swigCPtr, this);

}

public ParameterSpace() {

this(swigfaissJNI.new\_ParameterSpace(), true);

}

public long n\_combinations() {

return swigfaissJNI.ParameterSpace\_n\_combinations(swigCPtr, this);

}

public boolean combination\_ge(long c1, long c2) {

return swigfaissJNI.ParameterSpace\_combination\_ge(swigCPtr, this, c1, c2);

}

public String combination\_name(long cno) {

return swigfaissJNI.ParameterSpace\_combination\_name(swigCPtr, this, cno);

}

public void display() {

swigfaissJNI.ParameterSpace\_display(swigCPtr, this);

}

public ParameterRange add\_range(String name) {

return new ParameterRange(swigfaissJNI.ParameterSpace\_add\_range(swigCPtr, this, name), false);

}

public void initialize(Index index) {

swigfaissJNI.ParameterSpace\_initialize(swigCPtr, this, Index.getCPtr(index), index);

}

public void set\_index\_parameters(Index index, long cno) {

swigfaissJNI.ParameterSpace\_set\_index\_parameters\_\_SWIG\_0(swigCPtr, this, Index.getCPtr(index), index, cno);

}

public void set\_index\_parameters(Index index, String param\_string) {

swigfaissJNI.ParameterSpace\_set\_index\_parameters\_\_SWIG\_1(swigCPtr, this, Index.getCPtr(index), index, param\_string);

}

public void set\_index\_parameter(Index index, String name, double val) {

swigfaissJNI.ParameterSpace\_set\_index\_parameter(swigCPtr, this, Index.getCPtr(index), index, name, val);

}

public void update\_bounds(long cno, OperatingPoint op, SWIGTYPE\_p\_double upper\_bound\_perf, SWIGTYPE\_p\_double lower\_bound\_t) {

swigfaissJNI.ParameterSpace\_update\_bounds(swigCPtr, this, cno, OperatingPoint.getCPtr(op), op, SWIGTYPE\_p\_double.getCPtr(upper\_bound\_perf), SWIGTYPE\_p\_double.getCPtr(lower\_bound\_t));

}

public void explore(Index index, long nq, SWIGTYPE\_p\_float xq, AutoTuneCriterion crit, OperatingPoints ops) {

swigfaissJNI.ParameterSpace\_explore(swigCPtr, this, Index.getCPtr(index), index, nq, SWIGTYPE\_p\_float.getCPtr(xq), AutoTuneCriterion.getCPtr(crit), crit, OperatingPoints.getCPtr(ops), ops);

}

}