package modelisation;

import java.io.FileWriter;

import java.io.IOException;

import java.io.PrintWriter;

public class EcritureFichier

{

public static void main(String[] argv) throws IOException

{

/\*Le cube n’est pas complet\*/

/\*Chose à faire : /!\ Créer une boucle for pour implémenter les coordonnées des sommets \*/

PrintWriter test;

/\*On crée un fichier à partir de FileWriter() et on écrit sur ce fichier avec PrintWriter\*/

test = new PrintWriter(new FileWriter("fichier.stl"));

test.println("facet normal: 0 0 0");

test.println("outer loop");

test.println("vertex -0.500000 -0.500000 1.000000");

test.println("vertex 0.500000 -0.500000 1.000000");

test.println("vertex 0.500000 -0.500000 1.000000");

test.println("endloop");

test.println("enfacet");

test.println("facet normal: 0 0 0");

test.println("outer loop");

test.println("vertex -0.500000 -0.500000 1.000000");

test.println("vertex 0.500000 0.500000 1.000000");

test.println("vertex -0.500000 0.500000 1.000000");

test.println("endloop");

test.println("enfacet");

test.println("facet normal: 0 0 0");

test.println("outer loop");

test.println("vertex 0.500000 -0.500000 0.000000");

test.println("vertex -0.500000 -0.500000 0.000000");

test.println("vertex -0.500000 0.500000 0.000000");

test.println("endloop");

test.println("enfacet");

test.println("facet normal: 0 0 0");

test.println("outer loop");

test.println("vertex -0.500000 -0.500000 0.000000");

test.println("vertex -0.500000 -0.500000 1.000000");

test.println("vertex -0.500000 0.500000 1.000000");

test.println("endloop");

test.println("enfacet");

test.println("facet normal: 0 0 0");

test.println("outer loop");

test.println("vertex -0.500000 -0.500000 1.000000");

test.println("vertex -0.500000 0.500000 1.000000");

test.println("vertex -0.500000 0.500000 0.000000");

test.println("endloop");

test.println("enfacet");

test.println("facet normal: 0 0 0");

test.println("outer loop");

test.println("vertex 0.500000 -0.500000 1.000000");

test.println("vertex 0.500000 -0.500000 0.000000");

test.println("vertex 0.500000 0.500000 0.000000");

test.println("endloop");

test.println("enfacet");

test.println("facet normal: 0 0 0");

test.println("outer loop");

test.println("vertex 0.500000 -0.500000 1.000000");

test.println("vertex 0.500000 0.500000 0.000000");

test.println("vertex 0.500000 0.500000 1.000000");

test.println("endloop");

test.println("enfacet");

test.println("facet normal: 0 0 0");

test.println("outer loop");

test.println("vertex -0.500000 -0.500000 0.000000");

test.println("vertex 0.500000 -0.500000 1.000000");

test.println("vertex 0.500000 -0.500000 1.000000");

test.println("endloop");

test.println("enfacet");

test.println("facet normal: 0 0 0");

test.println("outer loop");

test.println("vertex -0.500000 -0.500000 1.000000");

test.println("vertex 0.500000 -0.500000 1.000000");

test.println("vertex 0.500000 -0.500000 1.000000");

test.println("endloop");

test.println("enfacet");

test.println("facet normal: 0 0 0");

test.println("outer loop");

test.println("vertex -0.500000 -0.500000 1.000000");

test.println("vertex 0.500000 -0.500000 1.000000");

test.println("vertex 0.500000 -0.500000 1.000000");

test.println("endloop");

test.println("enfacet");

test.println("facet normal: 0 0 0");

test.println("outer loop");

test.println("vertex -0.500000 -0.500000 1.000000");

test.println("vertex 0.500000 -0.500000 1.000000");

test.println("vertex 0.500000 -0.500000 1.000000");

test.println("endloop");

test.println("enfacet");

test.println("facet normal: 0 0 0");

test.println("outer loop");

test.println("vertex -0.500000 -0.500000 1.000000");

test.println("vertex 0.500000 -0.500000 1.000000");

test.println("vertex 0.500000 -0.500000 1.000000");

test.println("endloop");

test.println("enfacet");

/\*close() permet de fermer le flux et libère les ressources système qui lui sont associées. \*/

test.close();

}

}