**public class App {**

public static void main(String[] args) throws ClassNotFoundException, SQLException, FileNotFoundException, IOException {

XWPFDocument document = new XWPFDocument();

XWPFParagraph para = document.createParagraph();

para.setAlignment(ParagraphAlignment.CENTER);

XWPFRun run = para.createRun();

run.setBold(true);

run.setFontSize(36);

run.setText("Apache POI works well!");

XWPFParagraph para2 = document.createParagraph();

run = para2.createRun();

run.setText("\tApache POI is a Java library for working with MS Office documents."

+ " Apache POI has a mature interface to handle MS Excel files."

+ " The libraries for working with word and powerpoint files is sufficient, but evolving.");

//Creates a table

XWPFTable tab = document.createTable();

XWPFTableRow row = tab.getRow(0);

row.getCell(0).setText("Sl. No.");

row.addNewTableCell().setText("Name");

row.addNewTableCell().setText("Address");

row = tab.createRow();

row.getCell(0).setText("1.");

row.getCell(1).setText("Raman");

row.getCell(2).setText("Pondicherry");

document.write(new FileOutputStream("1.docx"));

}

}

**public class InputData {**

private int Dt;

private Angle dt;

private Angle Aon;

private int Rx;

private int Ry;

private int Rh;

private int VPx;

private int VPy;

private int VPh;

private int KSPx;

private int KSPy;

private int Dk;

private Angle AR;

private int Rn;

private String par;

private int p1;

private int p2;

private int p3;

private int d1;

private int d2;

private int d3;

private Angle s1;

private Angle s2;

private Angle s3;

private int deltaD;

private Angle deltad;

private int Dv;

private Angle dv;

private static Random random;

private double dxt;

private int p;

private Angle e;

private double kv;

private Angle kk;

private char rozm;

private Angle pz;

private double vd;

private char zar;

private Angle rv;

public InputData() {

random = new Random();

//Дт і dт

Dt = random.nextInt(5000) + 5000;

dt = new Angle(random.nextInt(6) - 3 + random.nextInt(100) / 100.);

//Аон

Aon = new Angle(random.nextInt(60));

if ((Aon.get() % 15 > 5)&&(Aon.get() % 15 <10)) {

Aon.set(Aon.get() + 7);

}

//R (x,y,h,n)

Rx = random.nextInt(10000) \* 10;

Ry = random.nextInt(10000) \* 10;

Rh = random.nextInt(150) + 80;

Rn = random.nextInt(20);

//пар.

par = random.nextInt(100) + "-" + random.nextInt(100) + "-" + random.nextInt(100);

//ВП (x,y,h)

VPx = Rx - (int) (Dt \* Math.cos((360 + dt.getDigr())\*Math.PI/180.)/5)\*5;

VPy = Ry - (int) (Dt \* Math.sin((360 + dt.getDigr())\*Math.PI/180.)/5)\*5;

VPh = random.nextInt(140) - 70 + Rh;

//Дк, АR

Dk = random.nextInt(1000) + 2000;

AR = new Angle(Aon.get() + random.nextInt(600) / 100. - 3);

//КСП (x,y)

KSPx = Rx - (int) (Dk \* Math.cos((360 + AR.getDigr())\*Math.PI/180.)/5)\*5;

KSPy = Ry - (int) (Dk \* Math.sin((360 + AR.getDigr())\*Math.PI/180.)/5)\*5;

//p - дальність

//d - попроавки на дальність

//s - поправки на зміщення

p1 = (int) (Dt / 1000 - random.nextInt(2) - 1) \* 1000;

p3 = (int) (Dt / 1000 + random.nextInt(2) + 2) \* 1000;

p2 = (int) ((p1 + p3) / 2000) \* 1000;

d1 = random.nextInt(10) \* 5 + 100;

d2 = d1 + random.nextInt(10) \* 5 + 20;

d3 = d2 + random.nextInt(10) \* 5 + 20;

s1 = new Angle(-1. \* (random.nextInt(10) / 100.));

s2 = new Angle(s1.get() - (random.nextInt(4) + 4) / 100.);

s3 = new Angle(s2.get() - (random.nextInt(4) + 4) / 100.);

//ΔД, Δd

int daln1, daln2;

int pd1, pd2;

Angle pa1, pa2;

double k, b;

if (Dt < p2) {

daln1 = p1;

daln2 = p2;

pd1 = d1;

pd2 = d2;

pa1 = new Angle(s1);

pa2 = new Angle(s2);

} else {

daln1 = p2;

daln2 = p3;

pd1 = d2;

pd2 = d3;

pa1 = new Angle(s2);

pa2 = new Angle(s3);

}

k = (daln1 - daln2) / (pd1 - pd2);

b = daln2 - pd2 \* k;

deltaD = (int) ((Dt - b) / k);

k = (daln1 - daln2) / (pa1.get() - pa2.get());

b = daln2 - pa2.get() \* k;

deltad = new Angle((Dt - b) / k);

//Дв, dв

Dv = Dt + deltaD;

dv = new Angle(dt.get() + deltad.get());

zar = getZar(Dv);

dxt = getDxt(Dv);

p = getP(Dv);

kv = Math.round(10.0 \* Dk / Dt) / 10.0;

pz = new Angle(Math.round(10.0 \* (AR.get() - (Aon.get() + dt.get()))) / 10.0);

kk = new Angle((double) (pz.get() / (Dt / 100.0)));

e = new Angle(Math.round((Rh - VPh) / (Dt / 1000.)) / 100.);

vd = getVd(Dv);

rv = getRv();

k = (Rx-KSPx)/(Ry-KSPy);

b = KSPx-KSPy\*k;

double tx;

tx = (VPy-b)/k;

if(tx<VPx) rozm='r';

else rozm='l';

}

public int getLong(int x1, int y1, int x2, int y2) {

int dx = x1 - x2;

int dy = y1 - y2;

int lng = (int) Math.sqrt(dx \* dx + dy \* dy);

return lng;

}

public Angle getAngle(int x1, int y1, int x2, int y2) {

int dx = x1 - x2;

int dy = y1 - y2;

Angle angle = new Angle(Math.atan(dy / dx));

return angle;

}

public char getZar(int daln) {

int d = daln + 2000;

char rez;

if (d > 12800) {

rez = 'p';

} else if (d > 11600) {

rez = 'z';

} else if (d > 10000) {

rez = '1';

} else if (d > 8400) {

rez = '2';

} else if (d > 6400) {

rez = '3';

} else {

rez = '4';

}

return rez;

}

public double getDxt(int daln) {

int d1, d2;

d1 = ((int) (daln / 200)) \* 200;

d2 = d1 + 200;

double dxt1, dxt2;

dxt1 = dxt2 = 0;

try {

Class.forName("org.hsqldb.jdbcDriver");

} catch (ClassNotFoundException e) {

System.exit(1);

}

Connection connection = null;

try {

connection = DriverManager.getConnection(

"jdbc:hsqldb:file:db", "SA", "");

} catch (SQLException e) {

System.exit(1);

}

try {

Statement statement = connection.createStatement();

String query;

query = "SELECT dx FROM strilba WHERE ZAR= '" + zar + "' and dal = " + d1;

ResultSet resultSet = statement.executeQuery(query);

if (resultSet.next()) {

dxt1 = resultSet.getDouble(1);

}

query = "SELECT dx FROM strilba WHERE ZAR= '" + zar + "' and dal = " + d2;

resultSet = statement.executeQuery(query);

if (resultSet.next()) {

dxt2 = resultSet.getDouble(1);

}

query = "SHUTDOWN";

statement.execute(query);

connection.close();

} catch (SQLException e) {

e.printStackTrace();

}

double res = (daln - d1) \* (dxt2 - dxt1) / 200;

return res + dxt1;

}

public int getP(int daln) {

int d1, d2;

d1 = ((int) (daln / 200)) \* 200;

d2 = d1 + 200;

double p1, p2;

p1 = p2 = 0;

try {

Class.forName("org.hsqldb.jdbcDriver");

} catch (ClassNotFoundException e) {

System.exit(1);

}

Connection connection = null;

try {

connection = DriverManager.getConnection(

"jdbc:hsqldb:file:db", "SA", "");

} catch (SQLException e) {

System.exit(1);

}

try {

Statement statement = connection.createStatement();

String query;

query = "SELECT p FROM strilba WHERE ZAR= '" + zar + "' and dal = " + d1;

ResultSet resultSet = statement.executeQuery(query);

if (resultSet.next()) {

p1 = resultSet.getDouble(1);

}

query = "SELECT p FROM strilba WHERE ZAR= '" + zar + "' and dal = " + d2;

resultSet = statement.executeQuery(query);

if (resultSet.next()) {

p2 = resultSet.getDouble(1);

}

query = "SHUTDOWN";

statement.execute(query);

connection.close();

} catch (SQLException e) {

e.printStackTrace();

}

double res = (daln - d1) \* (p2 - p1) / 200;

return (int) (res + p1);

}

public double getVd(int daln) {

int d1, d2;

d1 = ((int) (daln / 200)) \* 200;

d2 = d1 + 200;

double vd1, vd2;

vd1 = vd2 = 0;

try {

Class.forName("org.hsqldb.jdbcDriver");

} catch (ClassNotFoundException e) {

System.exit(1);

}

Connection connection = null;

try {

connection = DriverManager.getConnection(

"jdbc:hsqldb:file:db", "SA", "");

} catch (SQLException e) {

System.exit(1);

}

try {

Statement statement = connection.createStatement();

String query;

query = "SELECT vd FROM strilba WHERE ZAR= '" + zar + "' and dal = " + d1;

ResultSet resultSet = statement.executeQuery(query);

if (resultSet.next()) {

vd1 = resultSet.getDouble(1);

}

query = "SELECT vd FROM strilba WHERE ZAR= '" + zar + "' and dal = " + d2;

resultSet = statement.executeQuery(query);

if (resultSet.next()) {

vd2 = resultSet.getDouble(1);

}

query = "SHUTDOWN";

statement.execute(query);

connection.close();

} catch (SQLException e) {

e.printStackTrace();

}

double res = (daln - d1) \* (vd2 - vd1) / 200;

return res + vd1;

}

public Angle getRv() {

try {

Class.forName("org.hsqldb.jdbcDriver");

} catch (ClassNotFoundException e) {

System.exit(1);

}

Connection connection = null;

try {

connection = DriverManager.getConnection(

"jdbc:hsqldb:file:db", "SA", "");

} catch (SQLException e) {

System.exit(1);

}

int z = 0;

char h;

if (Rh - VPh > 0) {

h = '+';

} else {

h = '-';

}

try {

Statement statement = connection.createStatement();

String query;

query = "SELECT z FROM popr WHERE ZAR= '" + zar + "' and p = " + p + "and e = " + e + "and h = '" + h + "'";

ResultSet resultSet = statement.executeQuery(query);

if (resultSet.next()) {

z = resultSet.getInt(1);

}

query = "SHUTDOWN";

statement.execute(query);

connection.close();

} catch (SQLException e) {

e.printStackTrace();

}

Angle rv = new Angle(e.get() + 30 + z / 100.);

return rv;

}

public double DXT() {

return dxt;

}

public int P() {

return p;

}

public Angle KK() {

return kk;

}

public double KV() {

return kv;

}

public int DV() {

return Dv;

}

public Angle DVA() {

return dv;

}

public Angle RV() {

return rv;

}

public char ROZM(){

return rozm;

}

public String toString() {

return "Дт = " + Dt + "\tdт = ОН" + dt.toZ() + "\nAон = " + Aon + "\tZar = " + zar + " \tΔХтис = " + dxt + "\nRx = " + Rx + "\tRy = " + Ry

+ "\tRh = " + Rh + "\nВПx = " + VPx + "\tВПy = " + VPy + "\tВПh = " + VPh

+ "\nКСПx = " + KSPx + "\tКСПy = " + KSPy + "\nДк = " + Dk + "\tAR = "

+ AR + "\tN = " + Rn + "\tpar = " + par + "\n\t" + p1 + "\t" + p2 + "\t" + p3

+ "\n\t" + d1 + "\t" + d2 + "\t" + d3 + "\n\t" + s1.toZ() + "\t" + s2.toZ() + "\t" + s3.toZ()

+ "\nΔД = " + deltaD + "\tΔd = " + deltad.toZ() + "\nДв = " + Dv + "\tdВ = " + dv.toZ() + "\nП = " + p + "\tПЗ = " + pz

+ "\nKв = " + kv + "\tКк = " + kk + "\tε = " + e + "\tВд = " + vd + "\nРв = " + rv;

}

}

**public class Spost {**

char dov;

int pod;

char p;

static Random random;

public Spost() {

random = new Random();

int znak = random.nextInt(2);

if (znak == 1) {

dov = 'r';

} else {

dov = 'l';

}

pod = random.nextInt(20) + 15;

znak = random.nextInt(2);

if (znak == 1) {

p = '+';

} else {

p = '-';

}

}

public Spost(int p1, int p2) {

random = new Random();

int znak = random.nextInt(2);

if (znak == 1) {

dov = 'r';

} else {

dov = 'l';

}

pod = random.nextInt(p2 - p1) + p1;

znak = random.nextInt(2);

if (znak == 1) {

p = '+';

} else {

p = '-';

}

}

public Spost(int n) {

if (n == 0) {

dov = ' ';

pod = 0;

p = '?';

} else {

random = new Random();

int znak = random.nextInt(2);

if (znak == 1) {

dov = 'r';

} else {

dov = 'l';

}

pod = random.nextInt(20 / n) + (int) (15 / n);

znak = random.nextInt(2);

if (znak == 1) {

p = '+';

} else {

p = '-';

}

}

}

public void reset(int n){

if (n == 0) {

dov = ' ';

pod = 0;

p = '?';

} else {

random = new Random();

int znak = random.nextInt(2);

if (znak == 1) {

dov = 'r';

} else {

dov = 'l';

}

pod = random.nextInt(20 / n) + (int) (15 / n);

znak = random.nextInt(2);

if (znak == 1) {

p = '+';

} else {

p = '-';

}

}

}

public char getDov() {

return dov;

}

public int getPod() {

return pod;

}

public char getP() {

return p;

}

public String toString() {

String res = "";

if (p == '?') {

return "?";

}

if (dov == 'r') {

res = "П ";

} else {

res = "Л ";

}

res += pod + ", " + p;

return res;

}

}

**public class Strilba {**

private InputData id;

private Spost sp;

private int p;

private Angle rv;

private Angle dov;

public Strilba() {

id = new InputData();

sp = new Spost();

p = id.P();

rv = id.RV();

dov = id.DVA();

}

public void setSp(Spost sp) {

this.sp = sp;

}

public InputData getID() {

return id;

}

public int getP() {

return p;

}

public Angle getRv() { return rv; }

public Angle getDov() {

return dov;

}

public void doCorrect(Spost sp, int d) {

int dp = (int) (d / id.DXT());

if (sp.getP() == '+') {

dp = -dp;

}

p += dp;

double ddov1 = (int) (d \* id.KK().get());

if (id.ROZM() == 'r') {

if (sp.getP() == '-') {

ddov1 = -ddov1;

}

} else if (sp.getP() == '+') {

ddov1 = -ddov1;

}

double ddov2 = sp.pod \* id.KV();

if (sp.getDov() == 'r') {

ddov2 = -ddov2;

}

dov = new Angle(dov.get() + (ddov1 + ddov2) / 100.);

}

@Override

public String toString() {

return "\nП = " + p + "\tРв = " + rv + "\tДов = ОН" + dov.toZ();

}

}

**public class Angle {**

private double ang;

public Angle() {

ang = 0;

}

public Angle(double a) {

a %= 60;

ang = ((int)(a\*100.))/100.;

}

public Angle(Angle a) {

ang = a.get();

}

public void set(double a) {

a %= 60;

ang = ((int)(a\*100.))/100.;

}

public double get() {

return ang;

}

public void setDigr(double dgr) {

dgr %= 360;

ang = dgr / 6.;

}

public double getDigr() {

return ang \* 6.;

}

public String toString() {

String res="";

int a1 = Math.abs((int)ang);

res += (int)a1;

int a2 = Math.abs((int)(ang\*100.))-a1\*100;

if(a2<10) res += "-0"+a2;

else res += "-"+a2;

return res;

}

public String toZ(){

String res="";

if(ang<0)

res = "-";

else

res = "+";

int a1 = Math.abs((int)ang);

res += (int)a1;

int a2 = Math.abs((int)(ang\*100.))-a1\*100;

if(a2<10) res += "-0"+a2;

else res += "-"+a2;

return res;

}

}

Дт = 9339 dт = ОН+0-66

Aон = 16-00 Zar = 1 ΔХтис = 12.0

Rx = 79680 Ry = 91820 Rh = 103

ВПx = 70365 ВПy = 91180 ВПh = 46

КСПx = 79310 КСПy = 89520

Дк = 2334 AR = 13-46 N = 1 par = 3-3-75

8000 10000 12000

145 205 265

-0-02 -0-07 -0-14

ΔД = 184 Δd = -0-05

Дв = 9523 dВ = +0-61

П = 426 ПЗ = 3-20

Kв = 0.2 Кк = 0-03 ε = 0-06 Вд = 21.0

Рв = 30-06

Дт = 6598 dт = ОН-1-00

Aон = 1-00 Zar = 2 ΔХтис = 14.0

Rx = 8280 Ry = 31590 Rh = 146

ВПx = 1720 ВПy = 32275 ВПh = 215

КСПx = 5835 КСПy = 31490

Дк = 2450 AR = 0-39 N = 1 par = 42-46-55

4000 6000 8000

100 150 180

-0-08 -0-15 -0-19

ΔД = 158 Δd = -0-16

Дв = 6756 dВ = -1-14

П = 315 ПЗ = 0-40

Kв = 0.4 Кк = 0-00 ε = 0-10 Вд = 15.0

Рв = 29-90

Дт = 5051 dт = ОН+1-62

Aон = 14-00 Zar = 3 ΔХтис = 13.92

Rx = 51410 Ry = 30220 Rh = 204

ВПx = 46435 ВПy = 29370 ВПh = 232

КСПx = 51235 КСПy = 27495

Дк = 2733 AR = 14-38 N = 13 par = 34-32-43

3000 5000 7000

100 165 190

-0-02 -0-08 -0-14

ΔД = 165 Δd = -0-08

Дв = 5216 dВ = +1-54

П = 288 ПЗ = 1-20

Kв = 0.5 Кк = 0-02 ε = 0-06 Вд = 29.08

Рв = 29-94

Дт = 8882 dт = ОН+2-66

Aон = 49-00 Zar = 1 ΔХтис = 13.925

Rx = 21070 Ry = 49180 Rh = 160

ВПx = 12535 ВПy = 46740 ВПh = 130

КСПx = 19655 КСПy = 50765

Дк = 2129 AR = 51-96 N = 14 par = 75-15-69

7000 9000 11000

110 135 165

-0-01 -0-06 -0-12

ΔД = 133 Δd = -0-05

Дв = 9015 dВ = +2-61

П = 387 ПЗ = 0-30

Kв = 0.2 Кк = 0-00 ε = 0-03 Вд = 20.0

Рв = 30-03

Дт = 6006 dт = ОН+1-90

Aон = 44-00 Zar = 3 ΔХтис = 11.195

Rx = 71750 Ry = 13500 Rh = 155

ВПx = 65865 ВПy = 12315 ВПh = 132

КСПx = 72445 КСПy = 16380

Дк = 2967 AR = 42-73 N = 10 par = 9-88-69

5000 6000 8000

135 155 205

-0-02 -0-06 -0-11

ΔД = 155 Δd = -0-06

Дв = 6161 dВ = +1-84

П = 362 ПЗ = 3-20

Kв = 0.5 Кк = 0-05 ε = 0-04 Вд = 36.61

Рв = 30-04

Дт = 7007 dт = ОН-0-72

Aон = 43-00 Zar = 2 ΔХтис = 13.0

Rx = 48280 Ry = 43920 Rh = 121

ВПx = 41295 ВПy = 44445 ВПh = 80

КСПx = 49335 КСПy = 46520

Дк = 2810 AR = 41-31 N = 11 par = 55-65-1

5000 7000 9000

100 120 165

-0-09 -0-16 -0-23

ΔД = 119 Δd = -0-16

Дв = 7126 dВ = -0-88

П = 340 ПЗ = 1-00

Kв = 0.4 Кк = 0-01 ε = 0-06 Вд = 16.0

Рв = 30-06

Дт = 8594 dт = ОН+2-90

Aон = 4-00 Zar = 1 ΔХтис = 14.09

Rx = 15660 Ry = 24720 Rh = 92

ВПx = 7460 ВПy = 22155 ВПh = 144

КСПx = 13355 КСПy = 23885

Дк = 2457 AR = 3-32 N = 5 par = 68-46-56

7000 9000 11000

145 200 220

-0-05 -0-12 -0-16

ΔД = 188 Δd = -0-10

Дв = 8782 dВ = +2-80

П = 370 ПЗ = 3-60

Kв = 0.3 Кк = 0-04 ε = 0-06 Вд = 19.0

Рв = 29-94

Дт = 6081 dт = ОН+2-60

Aон = 30-00 Zar = 3 ΔХтис = 11.0

Rx = 68610 Ry = 28640 Rh = 190

ВПx = 62755 ВПy = 27005 ВПh = 170

КСПx = 70655 КСПy = 29145

Дк = 2110 AR = 32-33 N = 12 par = 39-97-74

5000 7000 9000

110 170 205

-0-03 -0-10 -0-17

ΔД = 142 Δd = -0-06

Дв = 6223 dВ = +2-54

П = 368 ПЗ = 0-30

Kв = 0.3 Кк = 0-00 ε = 0-03 Вд = 37.23

Рв = 30-03

Дт = 9610 dт = ОН-1-27

Aон = 20-00 Zar = z ΔХтис = 15.0

Rx = 99770 Ry = 1340 Rh = 166

ВПx = 90245 ВПy = 2610 ВПh = 128

КСПx = 100985 КСПy = -1350

Дк = 2953 AR = 19-05 N = 5 par = 80-69-65

8000 9000 11000

110 130 180

-0-01 -0-06 -0-10

ΔД = 145 Δd = -0-07

Дв = 9755 dВ = -1-34

П = 355 ПЗ = 0-30

Kв = 0.3 Кк = 0-00 ε = 0-04 Вд = 21.775

Рв = 30-04