**Министерство образования Республики Беларусь**

**Учреждение образования**

**«Белорусский государственный университет информатики   
и радиоэлектроники»**

Кафедра интеллектуальных

информационных технологий

**Отчёт**

по лабораторной работе № 3

по курсу «Графический интерфейс интеллектуальных систем»

Выполнили:

студенты группы 821701

Булова М.И.

Титенков П.В

Минск 2011

**Цель:**

Изучить алгоритмы построения параметрических кривых, использую форму Эрмита, Безье и B-сплайн. Реализовать графический редактор, позволяющий построение параметрических кривых. Реализовать базовые функции матричных вычислений и применить их при построении параметрических кривых.

**Алгоритм построения кривой с использованием формы Эрмита:**

public ArrayList<Cell> nextStepAlg() {

ArrayList<Cell> cells = new ArrayList();

float[][] tt = new float[][]{new float[]{t \* t \* t, t \* t, t, 1f}};

float[][] out = Matrix.mult(tt, cur\_matrix);

cells.add(new Cell(round(out[0][0]), round(out[0][1]), Color.GREEN, " t = " + t));

t += dt;

if (t - dt \* 1.5 > 1) {

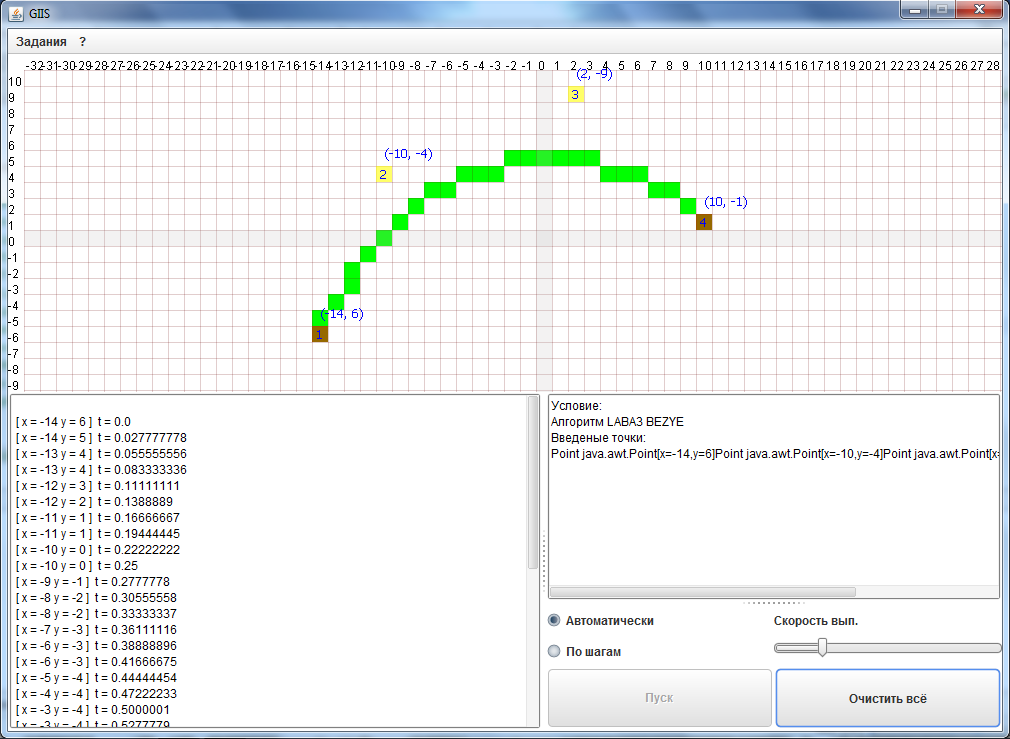
return null;

}

return cells;

}

**Пример работы программы:**



**Алгоритм построения кривой с использованием формы Безье:**

public ArrayList<Cell> nextStepAlg() {

ArrayList<Cell> cells = new ArrayList();

float[][] tt = new float[][]{new float[]{t \* t \* t, t \* t, t, 1f}};

float[][] out = Matrix.mult(tt, cur\_matrix);

cells.add(new Cell(round(out[0][0]), round(out[0][1]), Color.GREEN, " t = " + t));

t += dt;

if (t - dt \* 1.5 > 1) {

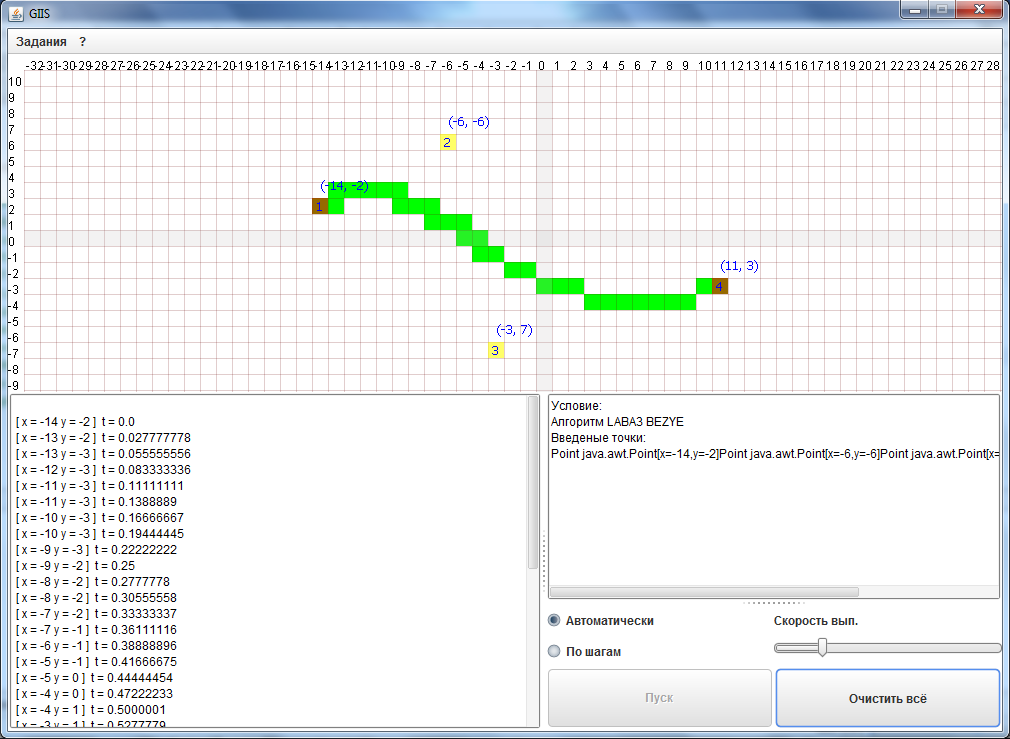
return null;

}

return cells;

}

**Пример работы программы:**



**Алгоритм построения кривой с использованием формы B-сплайн:**

public ArrayList<Cell> nextStepAlg() {

ArrayList<Cell> cells = new ArrayList();

if (iter < cp.size() - 2 && t - dt \* 1.5 >= 1) {

dx = Math.abs(cp.get((int) (iter - 1)).getPoint().x - cp.get((int) iter).getPoint().x);

dy = Math.abs(cp.get((int) (iter + 1)).getPoint().y - cp.get((int) (iter + 2)).getPoint().y);

t = 0;

dt = (float) (1f / (3 \* (dx + dy)));

data = new float[4][2];

for (int i = (int) iter; i < iter + 4; i++) {

data[i - (int) iter][0] = cp.get(i - 1).getPoint().x;

data[i - (int) iter][1] = cp.get(i - 1).getPoint().y;

}

cur\_matrix = Matrix.mult(B\_SPLINE, data);

iter++;

}

float[][] tt = new float[][]{new float[]{t \* t \* t, t \* t, t, 1}};

float[][] out = Matrix.mult(tt, cur\_matrix);

cells.add(new Cell(round(out[0][0] / 6), round(out[0][1] / 6), Color.GREEN, " t = " + t));

t += dt;

if (iter >= cp.size() - 2 && t - dt \* 1.5 >= 1) {

return null;

}

if (t - dt \* 1.5 >= 1) {

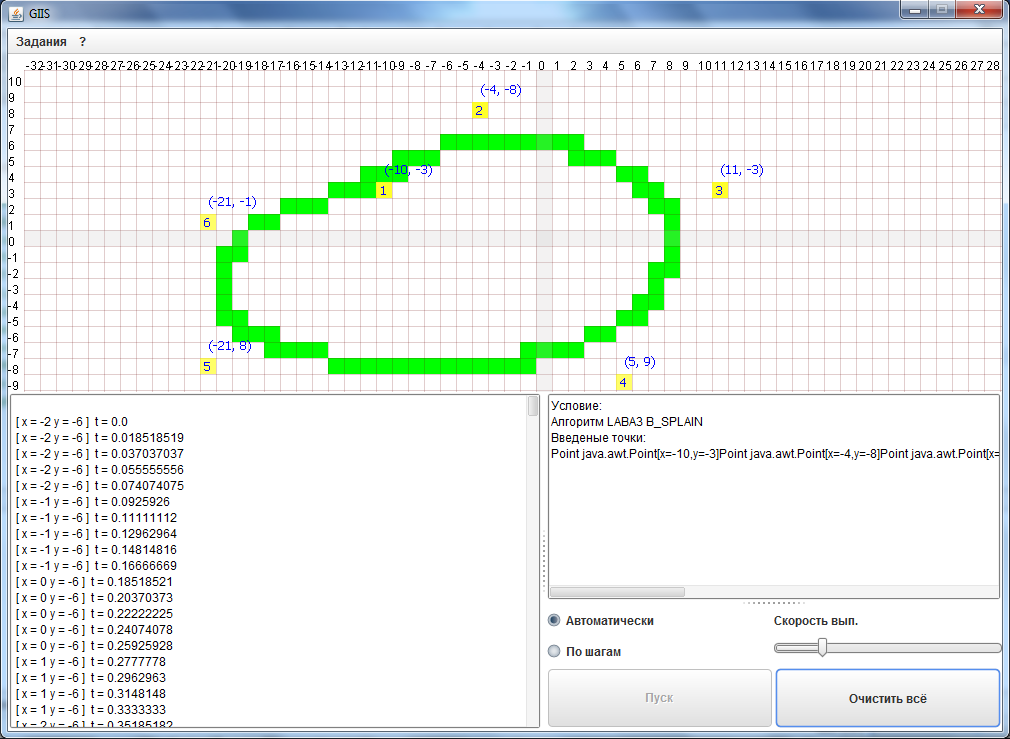
return nextStepAlg();

}

return cells;

}

**Пример работы программы:**



**Выводы:**

В ходе лабораторной работы были изучены принципы построения параметрических кривых, изучены формы Эрмита, Безье и B-сплайн. Были реализованы матрицы и алгоритмы операций над ними. Из приведенных выше алгоритмов наиболее оптимизированным под вычислительную технику является алгоритм Безье. B-сплайн обеспечивает построение более гладких кривых, чем другие методы сглаживания за счет того, что кривые не проходят точно через заданные точки. Главный недостаток алгоритма Эрмита – неявное задание векторов.