**Министерство образования Республики Беларусь**

**Учреждение образования**

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

Кафедра интеллектуальных

информационных технологий

**Отчёт**

по лабораторной работе № 5

по курсу «Графический интерфейс интеллектуальных систем»

Выполнили:

студенты группы 821701

Булова М.И.

Титенков П.В

Минск 2011

**Цель:**

Изучить алгоритмы заполнения замкнутых областей. Разработать редактор, позволяющий построение выпуклой области по множеству точек, а также позволяющий увидеть процесс заливки в пошаговом режиме.

**Алгоритм заполнения с затравкой простым способом:**

private boolean checkPoint(int x, int y){

for (Cell c : lines){

if (c.getX() == x && c.getY() == y){

return true;

}

}

for (Cell c : stack){

if (c.getX() == x && c.getY() == y){

stack.remove(c);

return false;

}

}

if (Math.abs(x \* 2) > PaintScreen.X\_SIZE - 2 || Math.abs(y \* 2) > PaintScreen.Y\_SIZE - 2){

return true;

}

return false;

}

/\*\*

\* Поиск следующей закрашиваемой точки.

\* @param cells список закрашиваемых точек

\*/

private void getNextPoint(ArrayList<Cell> cells){

if (stack.isEmpty()){

cells = null;

return;

}

Color color = new Color(0, 0, 255, 100);

Cell c = stack.get(stack.size() - 1);

cells.add(c);

stack.remove(c);

lines.add(c);

if (!checkPoint(c.getX() + 1, c.getY())){

stack.add(new Cell(c.getX() + 1, c.getY(), color, "x + 1, y"));

}

if (!checkPoint(c.getX(), c.getY() + 1)){

stack.add(new Cell(c.getX(), c.getY() + 1, color, "x, y + 1"));

}

if (!checkPoint(c.getX() - 1, c.getY())){

stack.add(new Cell(c.getX() - 1, c.getY(), color, "x - 1, y"));

}

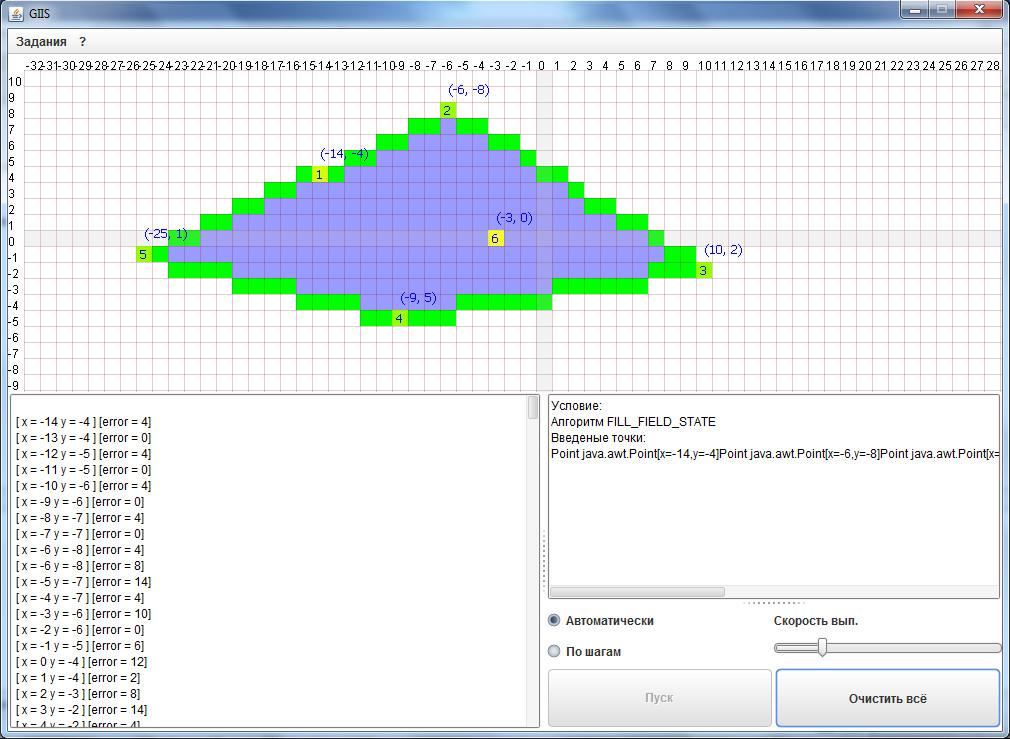
if (!checkPoint(c.getX(), c.getY() - 1)){

stack.add(new Cell(c.getX(), c.getY() - 1, color, "x, y - 1"));

}

}

**Пример работы программы:**



**Алгоритм заполнения с затравкой построчным способом:**

private boolean checkPoint(int x, int y) {

for (Cell c : lines) {

if (c.getX() == x && c.getY() == y) {

return true;

}

}

if (Math.abs(x \* 2) > PaintScreen.X\_SIZE - 2 || Math.abs(y \* 2) > PaintScreen.Y\_SIZE - 2) {

return true;

}

for (Cell cell : stack){

if (cell.getX() == x && cell.getY() == y){

stack.remove(cell);

return false;

}

}

return false;

}

private String flag = "";

/\*\*

\* Поиск следующей закрашиваемой точки.

\* @param cells список закрашиваемых точек

\*/

private void getNextPoint(ArrayList<Cell> cells) {

Color color = new Color(0, 0, 255, 100);

if (!flag.equals(FILL\_LINE\_STATE)) {

if (stack.isEmpty()) {

cells = null;

return;

}

Cell c = stack.get(stack.size() - 1);

cells.add(c);

stack.remove(c);

lines.add(c);

flag = FILL\_LINE\_STATE;

xmin = c.getX();

xmax = c.getX();

y = c.getY();

} else {

if (!checkPoint(xmax + 1, (int) y)) {

cells.add(new Cell(xmax + 1, (int) y, color, "xmax + 1, y"));

lines.add(new Cell(xmax + 1, (int) y, color, "xmax + 1, y"));

xmax++;

} else if (!checkPoint(xmin - 1, (int) y)) {

cells.add(new Cell(xmin - 1, (int) y, color, "xmin - 1, y"));

lines.add(new Cell(xmin - 1, (int) y, color, "xmin - 1, y"));

xmin--;

} else {

for (int i = xmin; i <= xmax; i++) {

if (!checkPoint(i, (int) y + 1)){

stack.add(new Cell(i, (int) y + 1, color, "x, y + 1"));

// break;

}

}

for (int i = xmin; i <= xmax; i++) {

if (!checkPoint(i, (int) y - 1)){

stack.add(new Cell(i, (int) y - 1, color, "x, y - 1"));

// break;

}

}

flag = "";

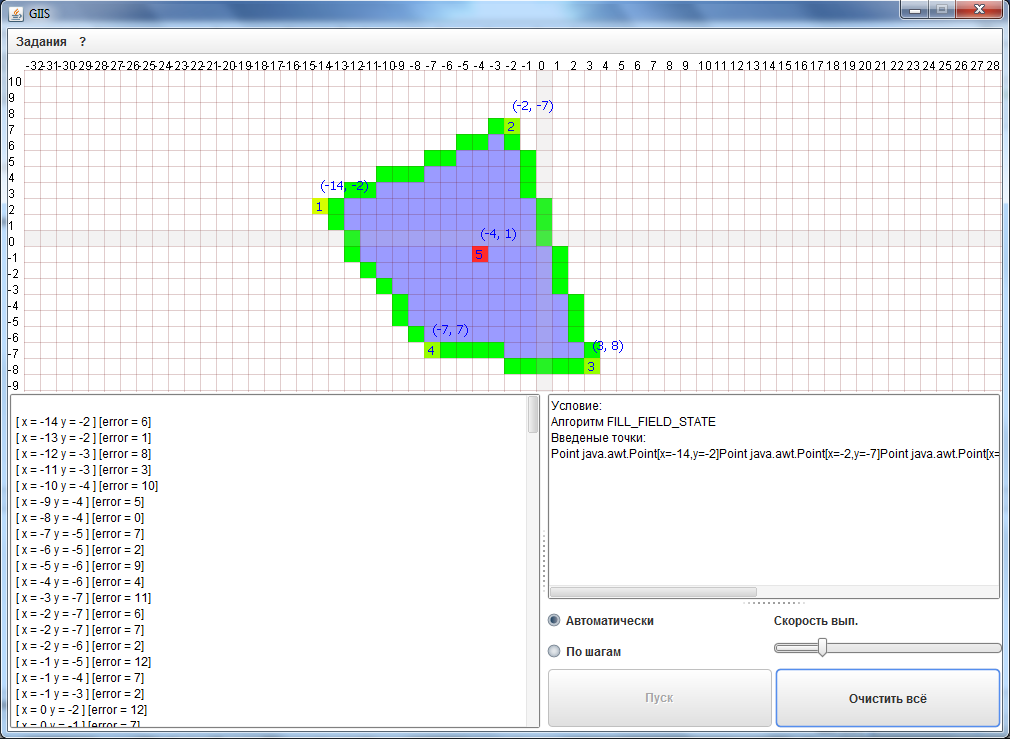
getNextPoint(cells);

}

}

}

**Пример работы программы:**



**Выводы:**

В ходе лабораторной работы были изучены принципы построения параметрических кривых, изучены формы Эрмита, Безье и B-сплайна. Были реализованы матрицы и алгоритмы операций над ними.