

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра САПР**

**ОТЧЕТ**  
**по лабораторной работе №2**  
**по дисциплине «Компьютерная графика»**  
**Тема: Формирование различных кривых с использованием**  
**ортогонального проектирования на плоскость визуализации**

Студенты гр. 8362

Преподаватель

\_\_\_\_\_  
\_\_\_\_\_

Ларионова Е.Е.

Матвеев Н.Д.

Матвеева И. В.

Санкт-Петербург

2021

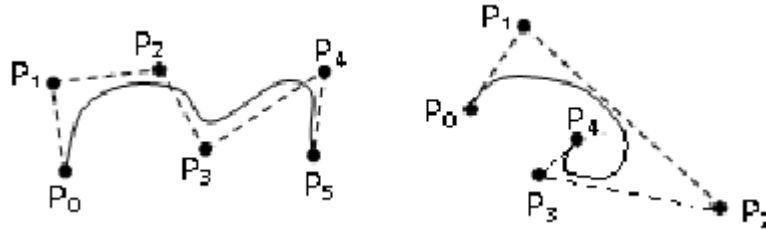
## **ЗАДАНИЕ**

Сформировать на плоскости кривую Безье на основе задающей ломаной, определяемой 3 и большим количеством точек. Обеспечить редактирование координат точек задающей ломаной с перерисовкой сплайна Безье.

## ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Кривые Безье. Их формирование и особенности.

Точки задания этих кривых только определяют ход кривой, сама строящаяся кривая в общем случае не проходит через внутренние точки задающего многоугольника:



Особенности:

1. Подходит по касательной к внешним ребрам (сторонам) задающего многоугольника, а остальные точки определяют ход кривой. Они позволяют качественно оценить ход кривой в зависимости от вида задающего многоугольника.
2. Кривая задается параметрически в функции от независимого параметра.
3. Это кривая  $n$ -ой степени, т.е. сколько ребер у задающего многоугольника – такой степени и получается кривая. Влиять на степень кривой можно только изменением количества задающих ее точек.

Математически такая кривая описывается параметрическим уравнением:

$$P(t) = \sum_{i=0}^n P_i \times N_{i,n}(t),$$

где  $P(t)$  – полиномиальная функция,

$P_i$  – координаты  $i$ -ой точки задания,

$N_{i,n}$  – весовой коэффициент  $i$ -той вершины (коэффициент бинома Ньютона),

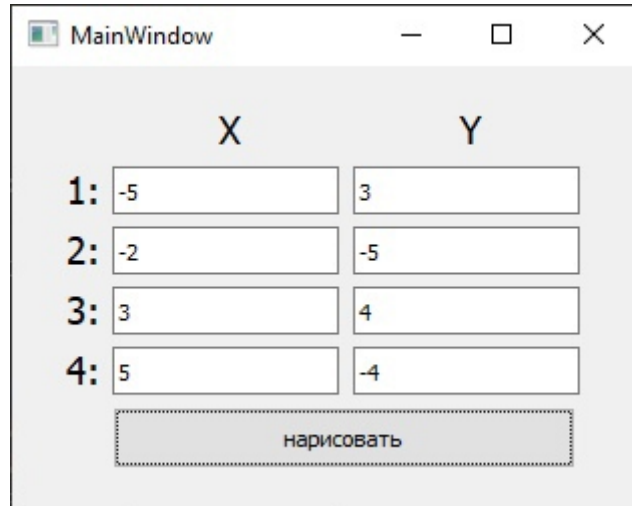
$i$  – номер точки,

$n$  – количество сторон задающего многоугольника

$t$  – задающий параметр, причем  $0 \leq t \leq 1$

## РЕАЛИЗАЦИЯ ПРОГРАММЫ

При запуске программы открываются 2 окна «MainWindow» и «Form». В «MainWindow» задаются координаты точек. В «Form» служит для отрисовки сплайна Безье (Рисунок 1 и Рисунок 2).



	X	Y
1:	-5	3
2:	-2	-5
3:	3	4
4:	5	-4

нарисовать

Рисунок 1 – Окно «MainWindow»

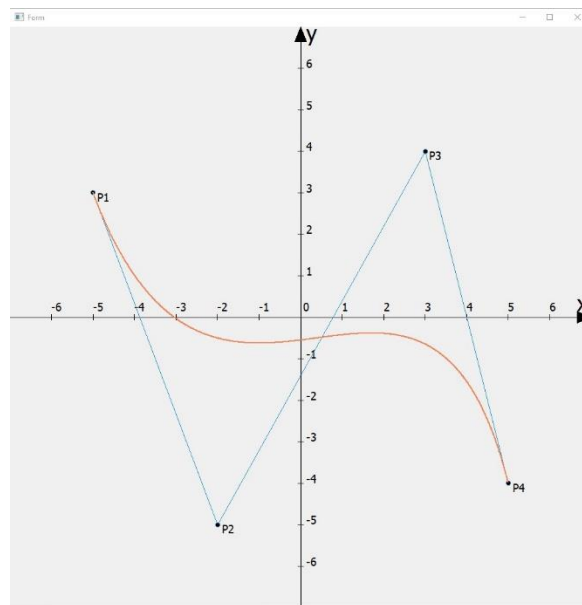


Рисунок 1 – Окно «Form»

## ПРИЛОЖЕНИЕ 1 – КОД ПРОГРАММЫ

### Файл main.cpp

```
#include <application.h>

int main(int argc, char *argv[])
{
    Application a(argc, argv);
    return a.exec();
}
```

### Файл mainwindow.cpp

```
#include "mainwindow.h"
#include "ui_mainwindow.h"

MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::on_pushButton_clicked()
{
    QPointF *p = new QPointF[4];
    p[0].setX(ui->lineEdit_x_1->text().toDouble());
```

```

    p[0].setY(ui->lineEdit_y_1->text().toDouble());
    p[1].setX(ui->lineEdit_x_2->text().toDouble());
    p[1].setY(ui->lineEdit_y_2->text().toDouble());
    p[2].setX(ui->lineEdit_x_3->text().toDouble());
    p[2].setY(ui->lineEdit_y_3->text().toDouble());
    p[3].setX(ui->lineEdit_x_4->text().toDouble());
    p[3].setY(ui->lineEdit_y_4->text().toDouble());
    emit send_control(p);
}

```

### Файл drawwindow.cpp

```

#include "mainwindow.h"
#include "ui_mainwindow.h"

MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::on_pushButton_clicked()
{
    QPointF *p = new QPointF[4];
    p[0].setX(ui->lineEdit_x_1->text().toDouble());

```

```

    p[0].setY(ui->lineEdit_y_1->text().toDouble());
    p[1].setX(ui->lineEdit_x_2->text().toDouble());
    p[1].setY(ui->lineEdit_y_2->text().toDouble());
    p[2].setX(ui->lineEdit_x_3->text().toDouble());
    p[2].setY(ui->lineEdit_y_3->text().toDouble());
    p[3].setX(ui->lineEdit_x_4->text().toDouble());
    p[3].setY(ui->lineEdit_y_4->text().toDouble());
    emit send_control(p);
}

```

### Файл application.cpp

```

#include "application.h"

#define T_STEP 0.001;

Application::Application(int argc, char *argv[])
    : QApplication(argc, argv)
{
    m = new MainWindow;
    m->show();
    d = new DrawWindow;
    d->show();
    connect(m, SIGNAL(send_control(QPointF*)),
            this, SLOT(recieve_control(QPointF*)));
    connect(this, SIGNAL(send_state(State*)),
            d, SLOT(recive_state(State*)));
}

void Application::recieve_control(QPointF* p)
{

```

```

    qreal step = T_STEP;
    State s;
    s.d = p;
    s.max = 5;
    for (size_t i = 0; i < 4; i++)
    {
        unsigned int tmp = static_cast<unsigned
int>(qAbs(p[i].x())>qAbs(p[i].y())?qAbs(p[i].x()):qAbs(
p[i].y()));
        s.max = s.max>tmp?s.max:tmp;
    }

    for (size_t i = 0; i*step < 1; i++)
    {
        s.p.push_back(sub_calc(p,i*step));
    }

    emit send_state(&s);
}

```

```

QPointF Application::sub_calc(QPointF* p, qreal t)
{
    return qPow(1-t,3)*p[0] + 3*qPow(1-t,2)*t*p[1] +
3*(1-t)*qPow(t,2)*p[2] + qPow(t,3)*p[3];
}

```