**Министерство образования Республики Беларусь**

**УО “Барановичский государственный университет”**

Инженерный факультет

Кафедра информационных технологий

И физико-математических дисциплин (ИСТ)

Средства визуальной разработки программных приложений

Лабораторная работа № 1

Вариант № 4

группа ИСТ-41

Ф.И.О. студента Борисов И.А.

Ф.И.О. преподавателя Шапович Е. Г.

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Барановичи, 2023

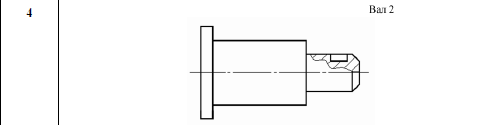
**Лабораторная работа 1**

**Графический интерфейс GDI+**

**Цель**: Овладение инструментальными средствами и методами графических объектно-ориентированных приложений.

**Задание 1**

Используя методы, определенных в классе Graphics, постройте на форме эскиз элемента технической системы согласно варианту



**Код программы:**

MainForm.cs

namespace WinGDITestApp

{

public partial class MainForm : Form

{

public string text;

public MainForm()

{

InitializeComponent();

}

// Run draw process

private void button1\_Click(object sender, EventArgs e)

{

Graphics g = Graphics.FromHwnd(this.Handle);

DrawCustomFigure drawCustom = new DrawCustomFigure();

g.SmoothingMode = System.Drawing.Drawing2D.SmoothingMode.AntiAlias;

g.Clear(Color.White);

drawCustom.drawFrame(g);

drawCustom.drawDashLine(g);

drawCustom.drawFlipFrame(g);

drawCustom.drawDottedLine(g);

);

}

private void MainForm\_MouseMove(object sender, MouseEventArgs e)

{

lbX.Text = Convert.ToString("X: " + e.Location.X);

lbY.Text = Convert.ToString("Y: " + e.Location.Y);

}

}

}

DrawCustomFigure.cs

// Drawing the frame of the figure

public void drawFrame(Graphics graphics)

{

graphics.DrawLines(m\_Pen, m\_PointsFrame);

// Lines

graphics.DrawLines(m\_Pen, m\_PointsLineIn);

graphics.DrawLines(m\_Pen, m\_PointsLineOut);

graphics.DrawLines(m\_Pen, m\_PointsLineFurther);

graphics.DrawLines(m\_Pen, m\_PointsLineInInside);

graphics.DrawLines(m\_Pen, m\_PointsLineU);

}

// Reflection of the drawn frame

public void drawFlipFrame(Graphics graphics)

{

int xmin = m\_PointsFrame.Min(x => x.X);

int xmax = m\_PointsFrame.Max(x => x.X);

graphics.TranslateTransform(0, xmin \* 8);

graphics.ScaleTransform(1, -1);

//graphics.TranslateTransform(0, xmin \* 2);

graphics.DrawLines(m\_Pen, m\_PointsFrame.ToArray());

}

// Drawing a dotted line

public void drawDottedLine(Graphics graphics)

{

// length, btw space, length space, end length

float[] dashValues = { 75, 10, 5, 20 };

Pen blackPen = new Pen(Color.Black, 1);

blackPen.DashPattern = dashValues;

graphics.DrawLine(blackPen, new Point(2, 200), new Point(600, 200));

}

// Drawing a dashed line

public void drawDashLine(Graphics graphics)

{

Pen dashPen = new Pen(Color.Red);

Pen curvePen = new Pen(Color.Blue);

// Set the width to 6.

dashPen.Width = 1.5F;

dashPen.DashStyle = System.Drawing.Drawing2D.DashStyle.Solid;

// Create a custom dash pattern.

// greenPen.DashPattern = new float[] { 4.0F, 2.0F, 1.0F, 3.0F };

// Drawing a curve

graphics.DrawCurve(curvePen, m\_PointsCurve);

// Drawing a line.

// array

graphics.DrawLine(dashPen, m\_PointsDash[0], m\_PointsDash[1]);

graphics.DrawLine(dashPen, m\_PointsDash[2], m\_PointsDash[3]);

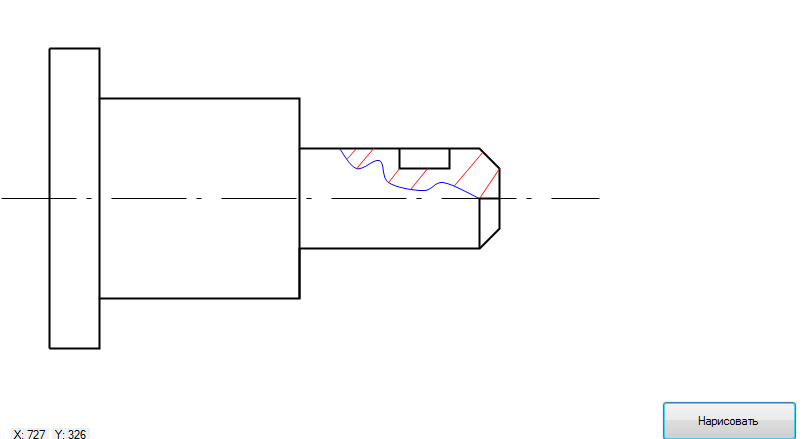
graphics.DrawLine(dashPen, m\_PointsDash[4], m\_PointsDash[5]);

graphics.DrawLine(dashPen, m\_PointsDash[6], m\_PointsDash[7]);

graphics.DrawLine(dashPen, m\_PointsDash[8], m\_PointsDash[9]);

graphics.DrawLine(dashPen, m\_PointsDash[10], m\_PointsDash[11]);}

**Результат программы:**

****

**Задание 2**

Разработать приложение, обладающее функциональностью, описанной в таблице. На форме приложения должна отображаться надпись с фамилией, инициалами, номером группы и 4 вариантом задания



**Код программы:**

MainForm.cs

namespace DrawLineApp{

public partial class MainForm : Form{

public MainForm(){

InitializeComponent();}

Point startPoint, endtPoint;

Point point1, point2;

bool drawing = false;

private void MainForm\_Paint(object sender, PaintEventArgs e){

var graphics = e.Graphics;

graphics.DrawString("Борисов И. А. ИСТ-41, 4 вариант", new Font("Arial", 14), Brushes.Blue, new Point(10, 10));

graphics.DrawLine(new Pen(Brushes.Green, 4), point1, point2);

}

private void MainForm\_MouseDown(object sender, MouseEventArgs e){

if (e.Button == MouseButtons.Right) {

this.drawing = true;

this.startPoint = new Point(400, 100); //e.x

this.endtPoint = new Point(400, 250);}}

private void MainForm\_MouseMove(object sender, MouseEventArgs e){

if (this.drawing){

this.point1 = new Point(this.startPoint.X, this.startPoint.Y);

this.point2 = new Point(this.endtPoint.X, this.endtPoint.Y);

Invalidate();} }

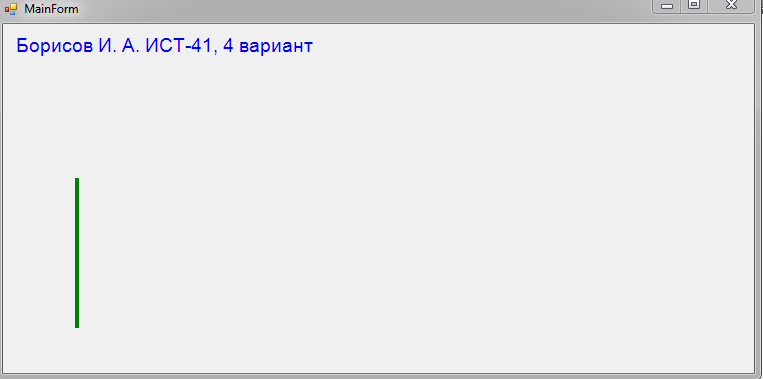
private void MainForm\_MouseUp(object sender, MouseEventArgs e){

this.drawing = false;}

private void MainForm\_KeyDown(object sender, KeyEventArgs e){

switch (e.KeyCode){

case Keys.T:{ // Up

 this.point1.Y -= 2;

this.point2.Y -= 2;

break;}

case Keys.F:{

this.point1.Y += 2;

this.point2.Y += 2;

break;}

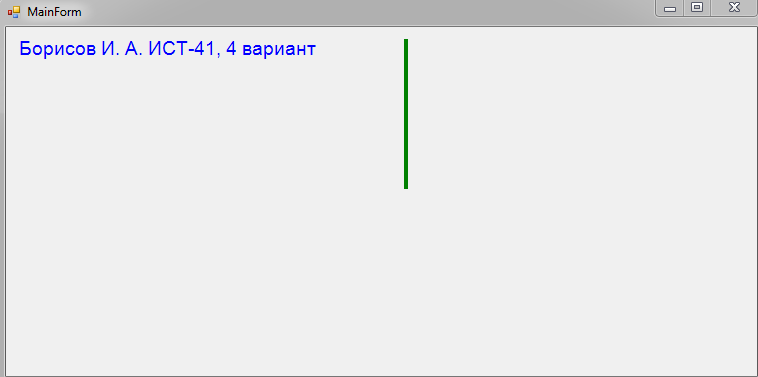
case Keys.G:{ // Left

if (e.Shift){

this.point1.X -= 2;

this.point2.X -= 2;}

break;}

 case Keys.H:{

if (e.Shift){

this.point1.X += 2;

this.point2.X += 2;}

break;}

case Keys.X:{

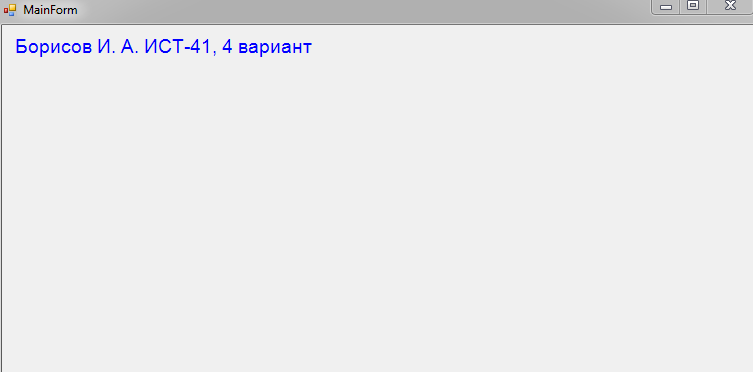
this.point1 = new Point(0, 0);

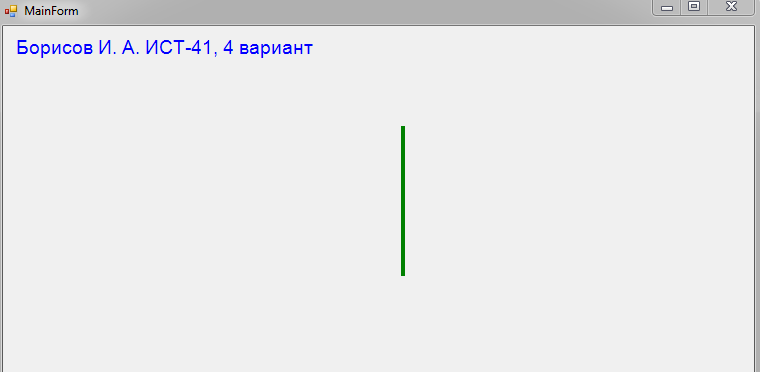
this.point2 = new Point(0, 0);

break;}}

Invalidate();}}}

**Результат программы:**





**Задание 3**

Разработать приложение для создания графика функции. Значение параметра a и начало и конец отрезка вводятся с клавиатуры. При изменении размера формы график должен перерисовываться в новом масштабе.

****

**Код программы:**

MainForm.cs

namespace GraphicApp{

public partial class MainForm : Form{

private float m\_A;

private int m\_N;

private int m\_Xmax;

private int m\_Xmin;

private int m\_Ymax;

private int m\_Ymin;

public MainForm(){

InitializeComponent();}

private void MainForm\_Resize(object sender, EventArgs e){

if (this.Visible){

groupData.Left = Width - this.groupData.Width - 20;

setAxisParameters();

Invalidate();}}

private void setAxisParameters()

{

m\_Xmax = int.Parse(tbXMax.Text);

m\_Xmin = int.Parse(tbXMin.Text);

m\_Ymax = int.Parse(tbYMax.Text);

m\_Ymin = int.Parse(tbYMin.Text);

m\_A = float.Parse(tbA.Text);

m\_N = int.Parse(tbN.Text);}

private double function(double x){

m\_A = float.Parse(tbA.Text);

return (m\_A \* Math.Sqrt(x) + Math.Sqrt(Math.Log(m\_A \* x)) +

Math.Pow(m\_A \* x, 3) - Math.Pow(m\_A \* x, 1 / 10)); }

private void drawGraph(){

setAxisParameters();

GraphPane pane = graphicsz.GraphPane;

pane.CurveList.Clear();

PointPairList list = new PointPairList();

int resultsSize = (m\_Xmax - m\_Xmin) \* m\_N;

int index = 0;

for (float x = m\_Xmin; x < m\_Xmax && index < resultsSize; x += 1f / m\_N, index++){

list.Add(x, function(x));

}

LineItem myCurve = pane.AddCurve("MyGraph", list, Color.Blue, SymbolType.None);

graphicsz.AxisChange();

graphicsz.Invalidate();}

private void btnPaint\_Click(object sender, EventArgs e){

if (m\_Xmin <= 0 && m\_Xmax >= 0){

if (m\_Ymin <= 0 && m\_Ymax >= 0){

drawGraph();

}}

else{

MessageBox.Show("Draw graph error!");

}}

private void MainForm\_Load(object sender, EventArgs e){

graphicsz.GraphPane.Title.Text = "";

graphicsz.GraphPane.Legend.IsVisible = false;

graphicsz.GraphPane.XAxis.Title.Text = "X";

graphicsz.GraphPane.YAxis.Title.Text = "Y";}

private void btnClearFields\_Click(object sender, EventArgs e){

tbA.Text = "";

tbN.Text = "";

tbXMax.Text = "";

tbXMin.Text = "";

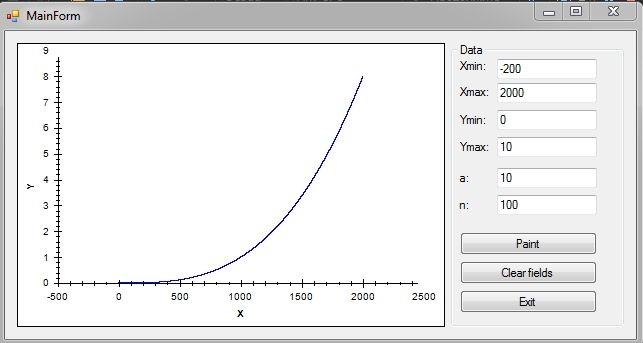
tbYMax.Text = "";

tbYMin.Text = "";}

private void btnExit\_Click(object sender, EventArgs e){

Application.Exit();}}}

**Результат программы:**

****

**Вывод**: овладел инструментальными средствами и методами графических объектно-ориентированных приложений.