**THE MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE**

**State University of Intelligent Technologies and Communications**

**Department of Software Engineering**

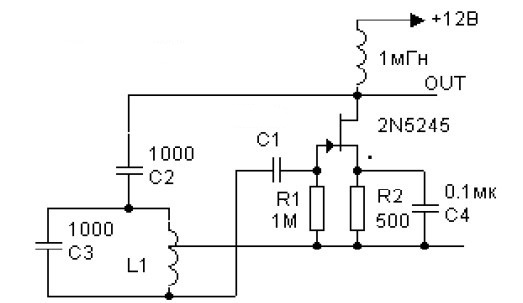
**Laboratory work №2**

**Variant №8**

**Made by Kadian Richard**

# Task

Draw the electrical chain according to the variant:



# Code

Resistor class:

using System.Drawing;

namespace Lab2 {

internal class Resistor : Element {

public Resistor(int x, int y, int d) : base(x, y, d, 1) {

}

public override void draw(Graphics G, Pen p) {

if (direction == 1) { // horizontal

} else {

G.DrawLine(p, xpos, ypos, xpos, ypos + 15);

G.DrawRectangle(p, xpos - 5, ypos + 15, 10, 40);

G.DrawLine(p, xpos, ypos + 15 + 40, xpos, ypos + 55 + 15);

}

}

}

}

Inductor Class:

using System;

using System.Drawing;

namespace Lab2 {

internal class Inductor : Element {

public Inductor(int x, int y, int d) : base(x, y, d, 1) {

}

public override void draw(Graphics G, Pen p) {

if (direction == 1) { // horizontal

G.DrawLine(p, xpos, ypos, xpos + 11, ypos);

G.DrawArc(p, xpos + 11, ypos - 8, 15, 30, -45, -90);

G.DrawArc(p, xpos + 25, ypos - 8, 15, 30, -45, -90);

G.DrawArc(p, xpos + 39, ypos - 8, 15, 30, -45, -90);

G.DrawLine(p, xpos + 54, ypos, xpos + 65, ypos);

} else {

G.DrawLine(p, xpos, ypos, xpos, ypos + 11);

G.DrawArc(p, xpos - 22, ypos + 11, 30, 15, -45, 90);

G.DrawArc(p, xpos - 22, ypos + 25, 30, 15, -45, 90);

G.DrawArc(p, xpos - 22, ypos + 39, 30, 15, -45, 90);

G.DrawLine(p, xpos, ypos + 54, xpos, ypos + 65);

}

}

}

}

Capacitor Class:

using System;

using System.Drawing;

namespace Lab2 {

internal class Capacitor : Element {

public Capacitor(int x, int y, int d) : base(x, y, d, 1) {

}

public override void draw(Graphics G, Pen p) {

if (direction == 1) {

G.DrawLine(p, xpos, ypos - 15, xpos, ypos + 15);

G.DrawLine(p, xpos + 5, ypos - 15, xpos + 5, ypos + 15);

} else {

G.DrawLine(p, xpos - 15, ypos, xpos + 15, ypos);

G.DrawLine(p, xpos - 15, ypos + 5, xpos + 15, ypos + 5);

}

}

}

}

Connector Class:

using System.Drawing;

namespace Lab2 {

internal class Connector : Element {

public Connector(int x, int y, int s) : base(x, y, 0, s) {

}

public override void draw(Graphics G, Pen p) {

G.DrawEllipse(p, xpos - (size / 2), ypos - (size / 2), size, size);

G.FillEllipse(new SolidBrush(Color.Black), xpos - (size / 2), ypos - (size / 2), size, size);

}

}

}

Arrow Class

using System.Drawing;

namespace Lab2 {

internal class Arrow : Element {

public Arrow(int x, int y, int d, int s) : base(x, y, d, s) {

}

public override void draw(Graphics G, Pen p) {

if (direction == 1) {

SolidBrush blackBrush = new SolidBrush(Color.Black);

Point a = new Point(xpos, ypos - 8);

Point b = new Point(xpos + size, ypos);

Point c = new Point(xpos, ypos + 8);

Point[] curvePoints = { a, b, c };

G.FillPolygon(blackBrush, curvePoints);

} else {

}

}

}

}

Line Class:

using System.Drawing;

namespace Lab2 {

public class Line : Element {

public Line(int x, int y, int d, int s) : base(x, y, d, s) {

}

public override void draw(Graphics G, Pen p) {

if (direction == 1) {

G.DrawLine(p, xpos, ypos, xpos + size, ypos);

} else {

G.DrawLine(p, xpos, ypos, xpos, ypos + size);

}

}

}

}

Element Class:

using System.Drawing;

namespace Lab2 {

public abstract class Element {

protected int xpos;

protected int ypos;

protected int direction;

protected int size;

public Element(int x, int y, int d, int s) {

xpos = x;

ypos = y;

if (d == 0)

direction = 0;

else

direction = 1;

size = s;

}

public abstract void draw(Graphics G, Pen p);

}

}

Form1 Class:

using System;

using System.Drawing;

using System.Windows.Forms;

namespace Lab2 {

public partial class Form1 : Form {

public Form1() {

InitializeComponent();

}

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

Graphics G = CreateGraphics();

Pen black = new Pen(Color.Black);

SolidBrush blackBrush = new SolidBrush(Color.Black);

int X0 = 150, Y0 = 300;

Element[] el = new Element[50];

el[0] = new Line(X0, Y0, 0, 30);

el[1] = new Capacitor(X0, Y0 + 30, 0); // C3

el[2] = new Line(X0, Y0 + 30 + 5, 0, 40);

el[3] = new Line(X0, Y0, 1, 60);

el[4] = new Connector(X0 + 60, Y0, 8);

el[5] = new Line(X0, Y0 + 30 + 5 + 40, 1, 100);

el[6] = new Connector(X0 + 100, Y0 + 30 + 5 + 40, 8);

el[7] = new Line(X0 + 60, Y0, 1, 40);

el[8] = new Line(X0 + 100, Y0, 0, 10);

el[9] = new Inductor(X0 + 100, Y0 + 10, 0); // L1

el[10] = new Line(X0 + 100, Y0 + 10 + 25, 1, 150);

el[11] = new Connector(X0 + 250, Y0 + 35, 8);

el[12] = new Line(X0 + 250, Y0 + 35, 1, 50);

el[13] = new Connector(X0 + 300, Y0 + 35, 8);

el[14] = new Line(X0 + 300, Y0 + 35, 1, 80);

el[15] = new Connector(X0 + 380, Y0 + 35, 8);

el[16] = new Line(X0 + 380, Y0 + 35, 1, 50);

el[17] = new Resistor(X0 + 250, Y0 + 35 - 70, 0); // R1

el[18] = new Resistor(X0 + 300, Y0 + 35 - 70, 0); // R2

el[19] = new Line(X0 + 380, Y0 + 35, 0, -33);

el[20] = new Capacitor(X0 + 380, Y0 + 35 - 33 - 5, 0);

el[21] = new Line(X0 + 380, Y0 + 35 - 38, 0, -32);

el[22] = new Line(X0 + 100, Y0 + 30 + 5 + 40, 1, 70);

el[23] = new Line(X0 + 170, Y0 + 75, 0, -40 - 70);

el[24] = new Line(X0 + 170, Y0 + 75 - 110, 1, 37);

el[25] = new Capacitor(X0 + 170 + 37, Y0 - 35, 1);

el[26] = new Line(X0 + 207 + 5, Y0 - 35, 1, 38);

el[27] = new Connector(X0 + 212 + 38, Y0 - 35, 8);

el[28] = new Line(X0 + 250, Y0 - 35, 0, -30);

el[29] = new Line(X0 + 250, Y0 - 35 - 30, 1, 50);

el[30] = new Line(X0 + 300, Y0 - 35 - 30, 0, 30);

el[31] = new Connector(X0 + 300, Y0 - 35, 8);

el[32] = new Line(X0 + 300, Y0 - 35, 1, 80);

el[33] = new Line(X0 + 60, Y0, 0, -30);

el[34] = new Capacitor(X0 + 60, Y0 - 30 - 5, 0); //

el[35] = new Line(X0 + 60, Y0 - 35, 0, -85);

el[36] = new Line(X0 + 60, Y0 - 35 - 85, 1, 240);

el[37] = new Connector(X0 + 300, Y0 - 120, 8);

el[38] = new Line(X0 + 300, Y0 - 120, 0, 30);

el[39] = new Line(X0 + 300, Y0 - 120 + 30, 1, -20);

el[40] = new Line(X0 + 300 - 20, Y0 - 90, 0, 25);

el[41] = new Line(X0 + 300, Y0 - 120, 1, 130);

el[42] = new Inductor(X0 + 300, Y0 - 120 - 65, 0);

el[43] = new Line(X0 + 300, Y0 - 185, 1, 125);

el[44] = new Arrow(X0 + 260, Y0 - 65, 1, 8);

el[45] = new Arrow(X0 + 415, Y0 - 185, 1, 15);

for (int i = 0; i < 46; i++)

el[i].draw(G, black);

}

}

}

# Screenshots

