Отчёт

Лабораторная работа 6

Рудометов Глеб БАС-14

**Задание**: Разработать многопоточное приложение, позволяющие динамически управлять потоками, выводящими на экран круги произвольного радиуса и цвета.

Список «Поток:» содержит созданные в программе потоки. Изменение параметров «Приоритет:», «Радиус:» и «Цвет:» применяются к выбранному потоку. Список «Приоритет» позволяет указать приоритет потока

В поле «Радиус:» вводится радиус кругов, рисуемых текущим потоком. При щелчке на поле «Цвет:» выводится стандартное диалоговое окно для выбора цвета потока

Дополнительно на форме должны быть две командные кнопки (рис. 3): 1) при нажатии на кнопку «Создать поток» создается новый поток с параметрами по умолчанию, имя потока добавляется в список потоков; 2) нажатие кнопки «Завершить поток» уничтожает выбранный в списке поток и удаляет его из этого списка. Список «Приоритет главного потока:» позволяет задать приоритет потока формы.

**Текст программы:**

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Threading;

namespace lab6

{

public partial class Form1 : Form

{

ThreadPriority[] tp = {ThreadPriority.Lowest,ThreadPriority.BelowNormal,ThreadPriority.Normal,ThreadPriority.AboveNormal,ThreadPriority.Highest};

List<myThread> list = new List<myThread>();

int num = 0; //номер текущего потока и макс.кол-во

public Graphics g;

Color DefaultColor;

public Form1()

{

InitializeComponent();

g=panel1.CreateGraphics();

DefaultColor = pictureBox1.BackColor;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = 2; //приоритет текущего потока

comboBox1.SelectedIndex = 2; //приоритет главного потока

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

if (colorDialog1.ShowDialog() == DialogResult.OK)

{

pictureBox1.BackColor=colorDialog1.Color;

list[num].setColor(colorDialog1.Color);

}

}

private void label2\_Click(object sender, EventArgs e)

{

}

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

{

num++;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

list[num].name = "Поток " + (list.Count-1);

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

pictureBox1.BackColor = DefaultColor;

comboBox3.SelectedIndex = 2;

}

private void button2\_Click(object sender, EventArgs e)

{

list[num].stop();

if (list.Count>1)

{

comboBox2.Items.Remove(list[num].name);

list.Remove(list[num]);

if (num > 0)num--;

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void numericUpDown1\_ValueChanged(object sender, EventArgs e)

{

list[num].setSize((int)numericUpDown1.Value);

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

num = comboBox2.SelectedIndex;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

Thread.CurrentThread.Priority = tp[comboBox1.SelectedIndex];

}

private void comboBox3\_SelectedIndexChanged(object sender, EventArgs e)

{

list[num].setPriority(tp[comboBox3.SelectedIndex]);

}

}

public class myThread //класс потока

{

public int size;

public Color color;

private Thread thread;

public ThreadPriority priority;

private Graphics g;

private void loop() {

Random r=new Random((int)(new DateTime()).Ticks);

while (true) {

lock (g)

{

g.FillEllipse((Brush)new SolidBrush(color), r.Next(0, 430 - size), r.Next(0, 415 - size), size, size);

//синхронизация потоков

//Thread.Sleep(1); //чтобы дать форме с низким приоритетом хоть какой то шанс среагировать !!!!!!

}

}

} //главный цикл потока

public myThread(string name,Graphics g,int size,Color color,ThreadPriority initial) {

this.name = name;

this.size = size;

this.color = color;

this.priority = initial;

this.g = g;

thread = new Thread(new ThreadStart(loop));

thread.Start();

} //создание потока

public string name;

public void setColor(Color a) { color = a; }

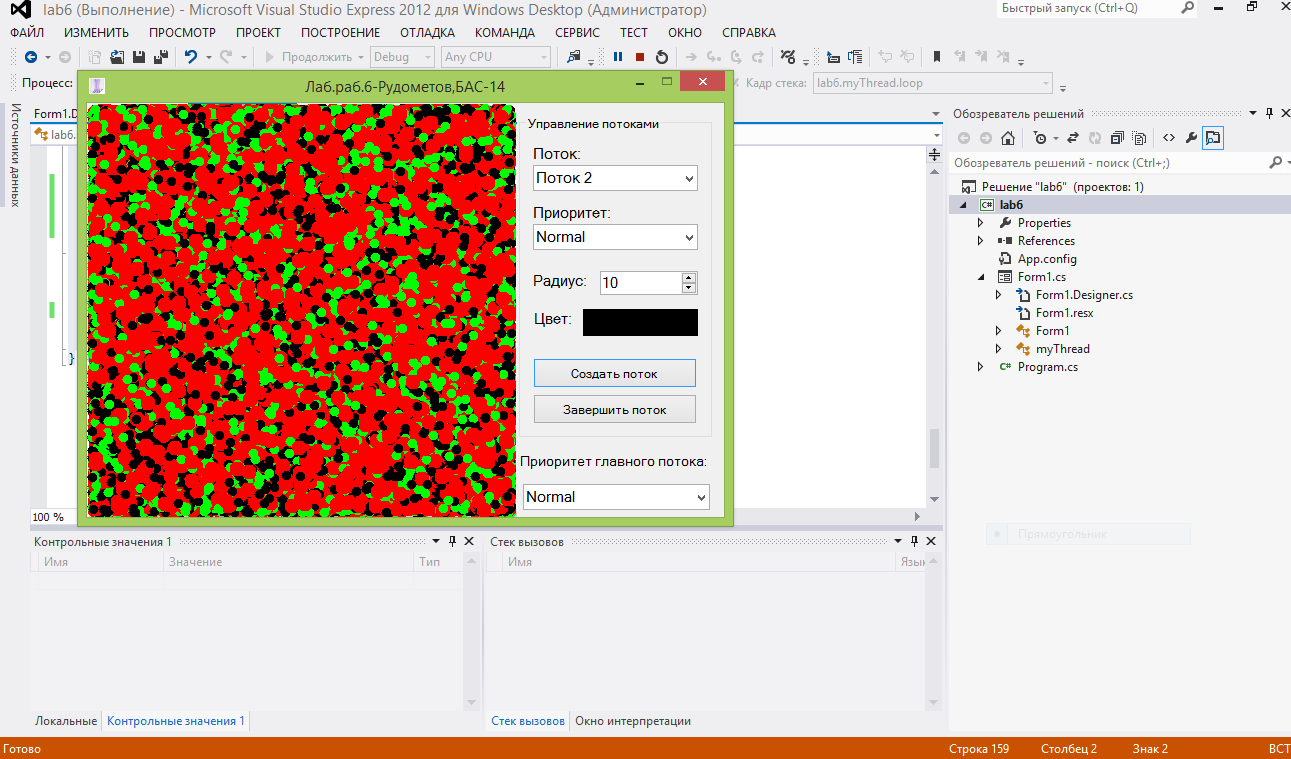
public void setSize(int s) { size = s; }

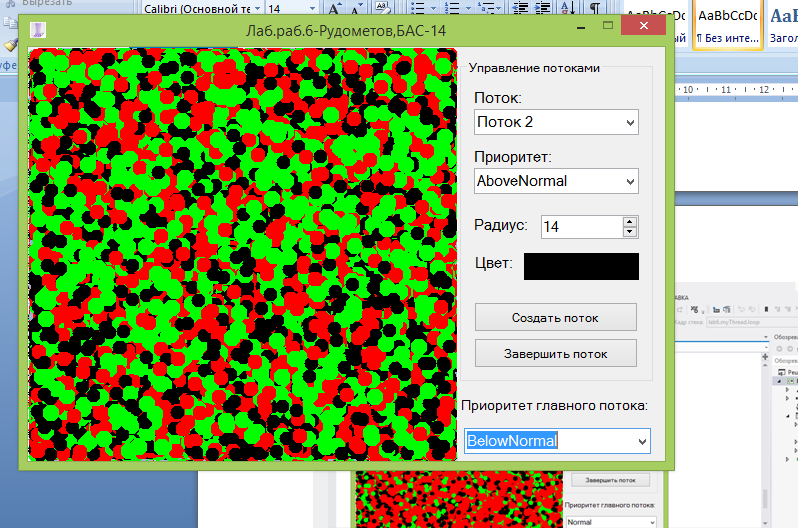
public void setPriority(ThreadPriority t) { priority = t; }

public void stop() { thread.Abort(); }

}

}





Дополнительная информация:

Отчёт

Лабораторная работа 6

Рудометов Глеб БАС-14

**Задание**: Разработать многопоточное приложение, позволяющие динамически управлять потоками, выводящими на экран круги произвольного радиуса и цвета.

Список «Поток:» содержит созданные в программе потоки. Изменение параметров «Приоритет:», «Радиус:» и «Цвет:» применяются к выбранному потоку. Список «Приоритет» позволяет указать приоритет потока

В поле «Радиус:» вводится радиус кругов, рисуемых текущим потоком. При щелчке на поле «Цвет:» выводится стандартное диалоговое окно для выбора цвета потока

Дополнительно на форме должны быть две командные кнопки (рис. 3): 1) при нажатии на кнопку «Создать поток» создается новый поток с параметрами по умолчанию, имя потока добавляется в список потоков; 2) нажатие кнопки «Завершить поток» уничтожает выбранный в списке поток и удаляет его из этого списка. Список «Приоритет главного потока:» позволяет задать приоритет потока формы.

**Текст программы:**

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Threading;

namespace lab6

{

public partial class Form1 : Form

{

ThreadPriority[] tp = {ThreadPriority.Lowest,ThreadPriority.BelowNormal,ThreadPriority.Normal,ThreadPriority.AboveNormal,ThreadPriority.Highest};

List<myThread> list = new List<myThread>();

int num = 0; //номер текущего потока и макс.кол-во

public Graphics g;

Color DefaultColor;

public Form1()

{

InitializeComponent();

g=panel1.CreateGraphics();

DefaultColor = pictureBox1.BackColor;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = 2; //приоритет текущего потока

comboBox1.SelectedIndex = 2; //приоритет главного потока

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

if (colorDialog1.ShowDialog() == DialogResult.OK)

{

pictureBox1.BackColor=colorDialog1.Color;

list[num].setColor(colorDialog1.Color);

}

}

private void label2\_Click(object sender, EventArgs e)

{

}

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

{

num++;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

list[num].name = "Поток " + (list.Count-1);

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

pictureBox1.BackColor = DefaultColor;

comboBox3.SelectedIndex = 2;

}

private void button2\_Click(object sender, EventArgs e)

{

list[num].stop();

if (list.Count>1)

{

comboBox2.Items.Remove(list[num].name);

list.Remove(list[num]);

if (num > 0)num--;

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void numericUpDown1\_ValueChanged(object sender, EventArgs e)

{

list[num].setSize((int)numericUpDown1.Value);

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

num = comboBox2.SelectedIndex;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

Thread.CurrentThread.Priority = tp[comboBox1.SelectedIndex];

}

private void comboBox3\_SelectedIndexChanged(object sender, EventArgs e)

{

list[num].setPriority(tp[comboBox3.SelectedIndex]);

}

}

public class myThread //класс потока

{

public int size;

public Color color;

private Thread thread;

public ThreadPriority priority;

private Graphics g;

private void loop() {

Random r=new Random((int)(new DateTime()).Ticks);

while (true) {

lock (g)

{

g.FillEllipse((Brush)new SolidBrush(color), r.Next(0, 430 - size), r.Next(0, 415 - size), size, size);

//синхронизация потоков

//Thread.Sleep(1); //чтобы дать форме с низким приоритетом хоть какой то шанс среагировать !!!!!!

}

}

} //главный цикл потока

public myThread(string name,Graphics g,int size,Color color,ThreadPriority initial) {

this.name = name;

this.size = size;

this.color = color;

this.priority = initial;

this.g = g;

thread = new Thread(new ThreadStart(loop));

thread.Start();

} //создание потока

public string name;

public void setColor(Color a) { color = a; }

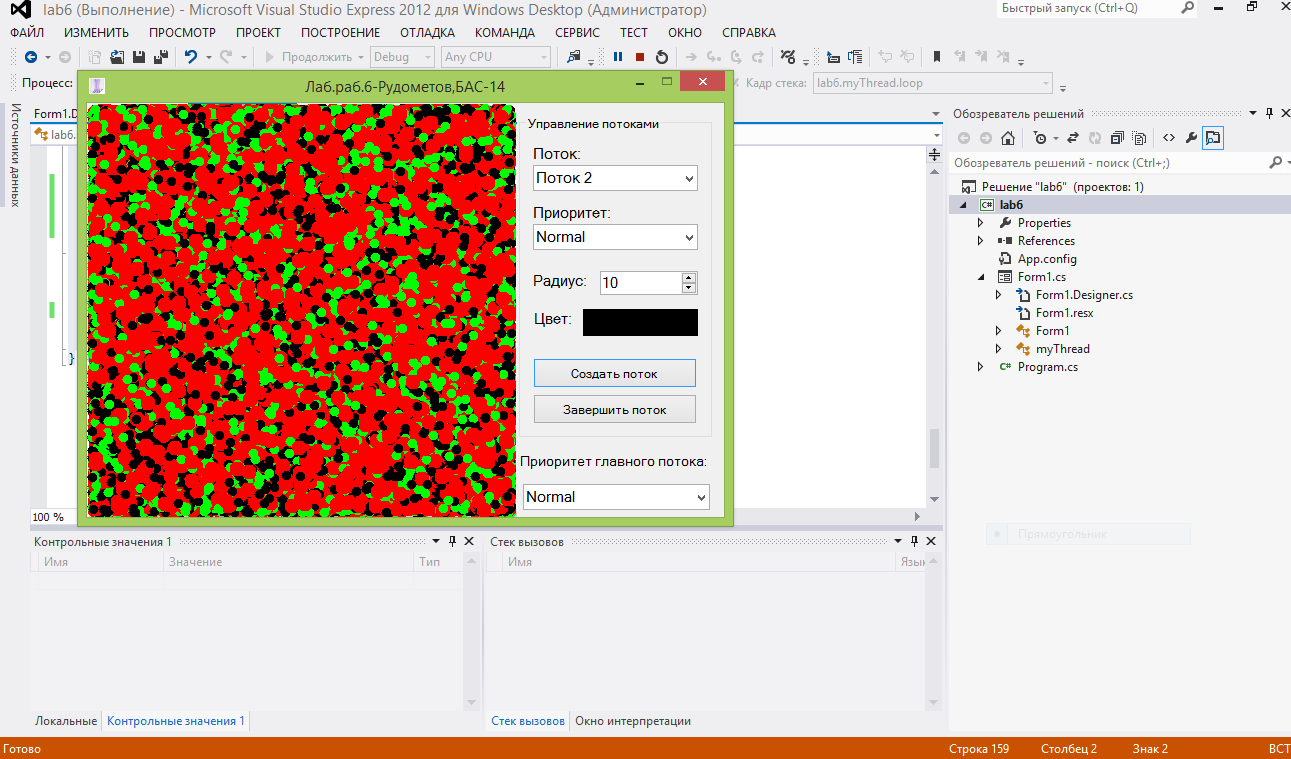
public void setSize(int s) { size = s; }

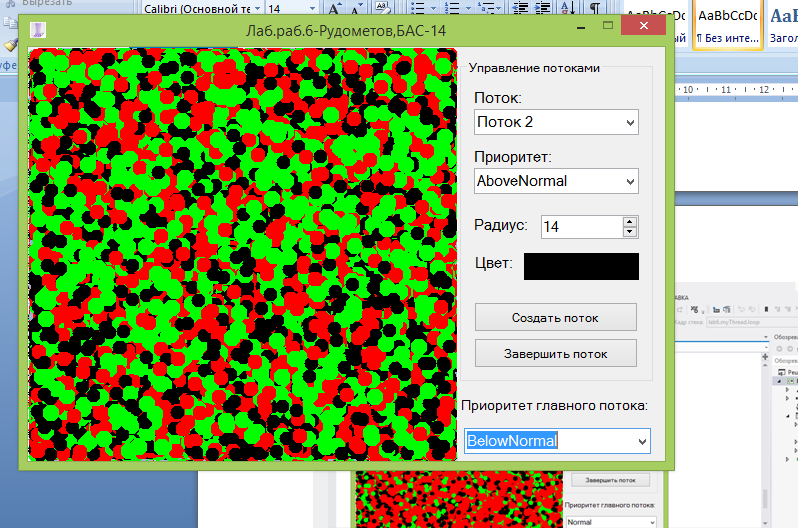
public void setPriority(ThreadPriority t) { priority = t; }

public void stop() { thread.Abort(); }

}

}





Отчёт

Лабораторная работа 6

Рудометов Глеб БАС-14

**Задание**: Разработать многопоточное приложение, позволяющие динамически управлять потоками, выводящими на экран круги произвольного радиуса и цвета.

Список «Поток:» содержит созданные в программе потоки. Изменение параметров «Приоритет:», «Радиус:» и «Цвет:» применяются к выбранному потоку. Список «Приоритет» позволяет указать приоритет потока

В поле «Радиус:» вводится радиус кругов, рисуемых текущим потоком. При щелчке на поле «Цвет:» выводится стандартное диалоговое окно для выбора цвета потока

Дополнительно на форме должны быть две командные кнопки (рис. 3): 1) при нажатии на кнопку «Создать поток» создается новый поток с параметрами по умолчанию, имя потока добавляется в список потоков; 2) нажатие кнопки «Завершить поток» уничтожает выбранный в списке поток и удаляет его из этого списка. Список «Приоритет главного потока:» позволяет задать приоритет потока формы.

**Текст программы:**

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Threading;

namespace lab6

{

public partial class Form1 : Form

{

ThreadPriority[] tp = {ThreadPriority.Lowest,ThreadPriority.BelowNormal,ThreadPriority.Normal,ThreadPriority.AboveNormal,ThreadPriority.Highest};

List<myThread> list = new List<myThread>();

int num = 0; //номер текущего потока и макс.кол-во

public Graphics g;

Color DefaultColor;

public Form1()

{

InitializeComponent();

g=panel1.CreateGraphics();

DefaultColor = pictureBox1.BackColor;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = 2; //приоритет текущего потока

comboBox1.SelectedIndex = 2; //приоритет главного потока

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

if (colorDialog1.ShowDialog() == DialogResult.OK)

{

pictureBox1.BackColor=colorDialog1.Color;

list[num].setColor(colorDialog1.Color);

}

}

private void label2\_Click(object sender, EventArgs e)

{

}

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

{

num++;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

list[num].name = "Поток " + (list.Count-1);

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

pictureBox1.BackColor = DefaultColor;

comboBox3.SelectedIndex = 2;

}

private void button2\_Click(object sender, EventArgs e)

{

list[num].stop();

if (list.Count>1)

{

comboBox2.Items.Remove(list[num].name);

list.Remove(list[num]);

if (num > 0)num--;

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void numericUpDown1\_ValueChanged(object sender, EventArgs e)

{

list[num].setSize((int)numericUpDown1.Value);

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

num = comboBox2.SelectedIndex;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

Thread.CurrentThread.Priority = tp[comboBox1.SelectedIndex];

}

private void comboBox3\_SelectedIndexChanged(object sender, EventArgs e)

{

list[num].setPriority(tp[comboBox3.SelectedIndex]);

}

}

public class myThread //класс потока

{

public int size;

public Color color;

private Thread thread;

public ThreadPriority priority;

private Graphics g;

private void loop() {

Random r=new Random((int)(new DateTime()).Ticks);

while (true) {

lock (g)

{

g.FillEllipse((Brush)new SolidBrush(color), r.Next(0, 430 - size), r.Next(0, 415 - size), size, size);

//синхронизация потоков

//Thread.Sleep(1); //чтобы дать форме с низким приоритетом хоть какой то шанс среагировать !!!!!!

}

}

} //главный цикл потока

public myThread(string name,Graphics g,int size,Color color,ThreadPriority initial) {

this.name = name;

this.size = size;

this.color = color;

this.priority = initial;

this.g = g;

thread = new Thread(new ThreadStart(loop));

thread.Start();

} //создание потока

public string name;

public void setColor(Color a) { color = a; }

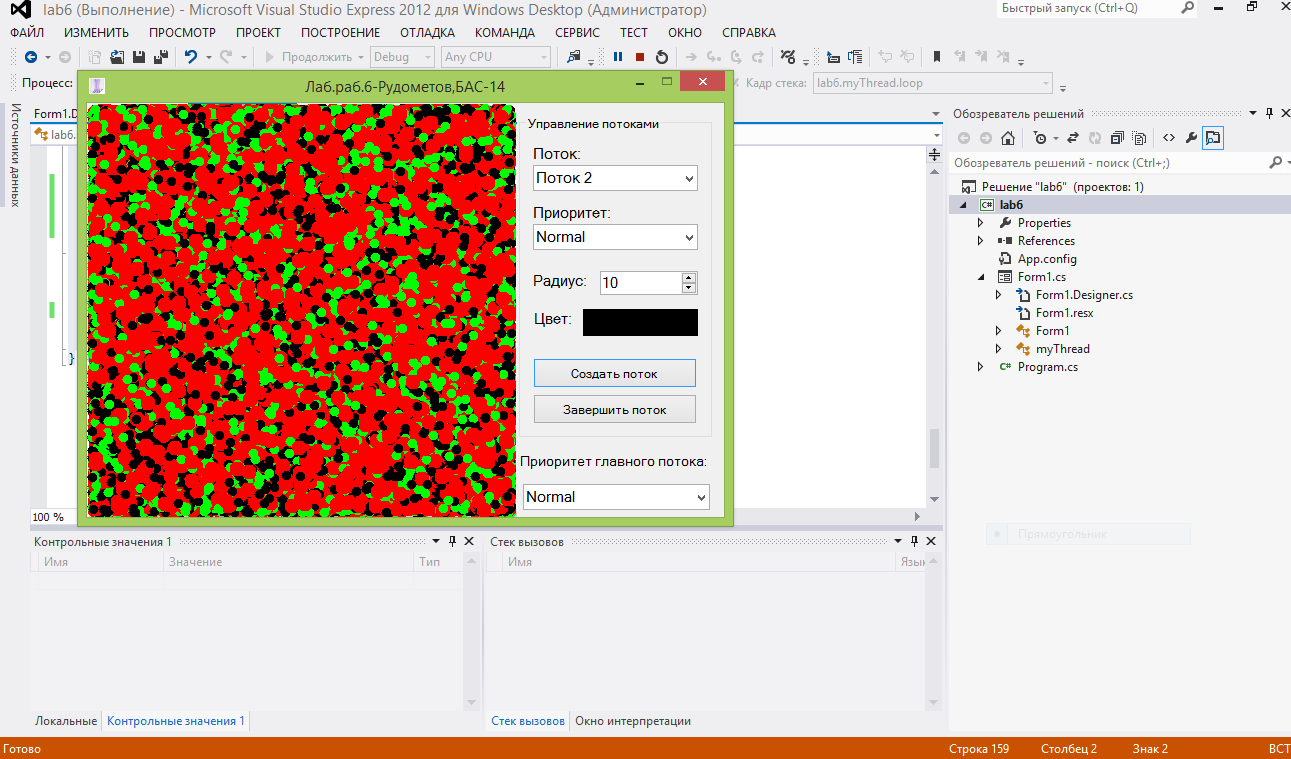
public void setSize(int s) { size = s; }

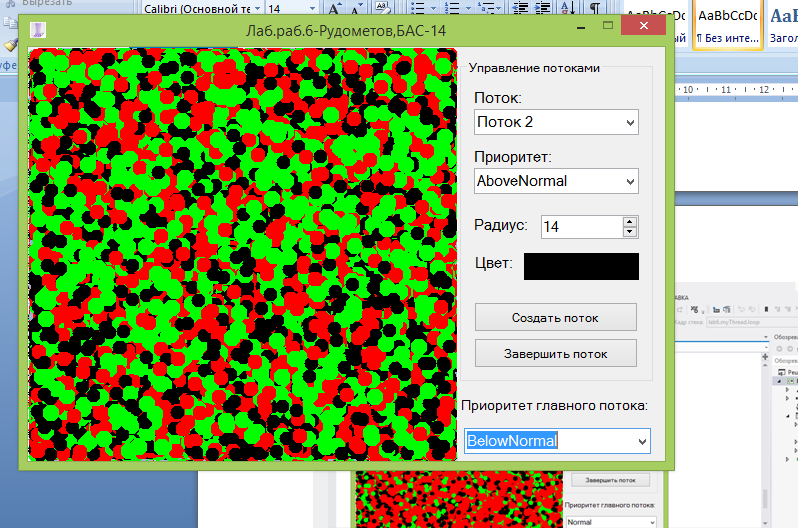
public void setPriority(ThreadPriority t) { priority = t; }

public void stop() { thread.Abort(); }

}

}





Дополнительная информация:

Отчёт

Лабораторная работа 6

Рудометов Глеб БАС-14

**Задание**: Разработать многопоточное приложение, позволяющие динамически управлять потоками, выводящими на экран круги произвольного радиуса и цвета.

Список «Поток:» содержит созданные в программе потоки. Изменение параметров «Приоритет:», «Радиус:» и «Цвет:» применяются к выбранному потоку. Список «Приоритет» позволяет указать приоритет потока

В поле «Радиус:» вводится радиус кругов, рисуемых текущим потоком. При щелчке на поле «Цвет:» выводится стандартное диалоговое окно для выбора цвета потока

Дополнительно на форме должны быть две командные кнопки (рис. 3): 1) при нажатии на кнопку «Создать поток» создается новый поток с параметрами по умолчанию, имя потока добавляется в список потоков; 2) нажатие кнопки «Завершить поток» уничтожает выбранный в списке поток и удаляет его из этого списка. Список «Приоритет главного потока:» позволяет задать приоритет потока формы.

**Текст программы:**

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Threading;

namespace lab6

{

public partial class Form1 : Form

{

ThreadPriority[] tp = {ThreadPriority.Lowest,ThreadPriority.BelowNormal,ThreadPriority.Normal,ThreadPriority.AboveNormal,ThreadPriority.Highest};

List<myThread> list = new List<myThread>();

int num = 0; //номер текущего потока и макс.кол-во

public Graphics g;

Color DefaultColor;

public Form1()

{

InitializeComponent();

g=panel1.CreateGraphics();

DefaultColor = pictureBox1.BackColor;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = 2; //приоритет текущего потока

comboBox1.SelectedIndex = 2; //приоритет главного потока

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

if (colorDialog1.ShowDialog() == DialogResult.OK)

{

pictureBox1.BackColor=colorDialog1.Color;

list[num].setColor(colorDialog1.Color);

}

}

private void label2\_Click(object sender, EventArgs e)

{

}

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

{

num++;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

list[num].name = "Поток " + (list.Count-1);

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

pictureBox1.BackColor = DefaultColor;

comboBox3.SelectedIndex = 2;

}

private void button2\_Click(object sender, EventArgs e)

{

list[num].stop();

if (list.Count>1)

{

comboBox2.Items.Remove(list[num].name);

list.Remove(list[num]);

if (num > 0)num--;

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void numericUpDown1\_ValueChanged(object sender, EventArgs e)

{

list[num].setSize((int)numericUpDown1.Value);

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

num = comboBox2.SelectedIndex;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

Thread.CurrentThread.Priority = tp[comboBox1.SelectedIndex];

}

private void comboBox3\_SelectedIndexChanged(object sender, EventArgs e)

{

list[num].setPriority(tp[comboBox3.SelectedIndex]);

}

}

public class myThread //класс потока

{

public int size;

public Color color;

private Thread thread;

public ThreadPriority priority;

private Graphics g;

private void loop() {

Random r=new Random((int)(new DateTime()).Ticks);

while (true) {

lock (g)

{

g.FillEllipse((Brush)new SolidBrush(color), r.Next(0, 430 - size), r.Next(0, 415 - size), size, size);

//синхронизация потоков

//Thread.Sleep(1); //чтобы дать форме с низким приоритетом хоть какой то шанс среагировать !!!!!!

}

}

} //главный цикл потока

public myThread(string name,Graphics g,int size,Color color,ThreadPriority initial) {

this.name = name;

this.size = size;

this.color = color;

this.priority = initial;

this.g = g;

thread = new Thread(new ThreadStart(loop));

thread.Start();

} //создание потока

public string name;

public void setColor(Color a) { color = a; }

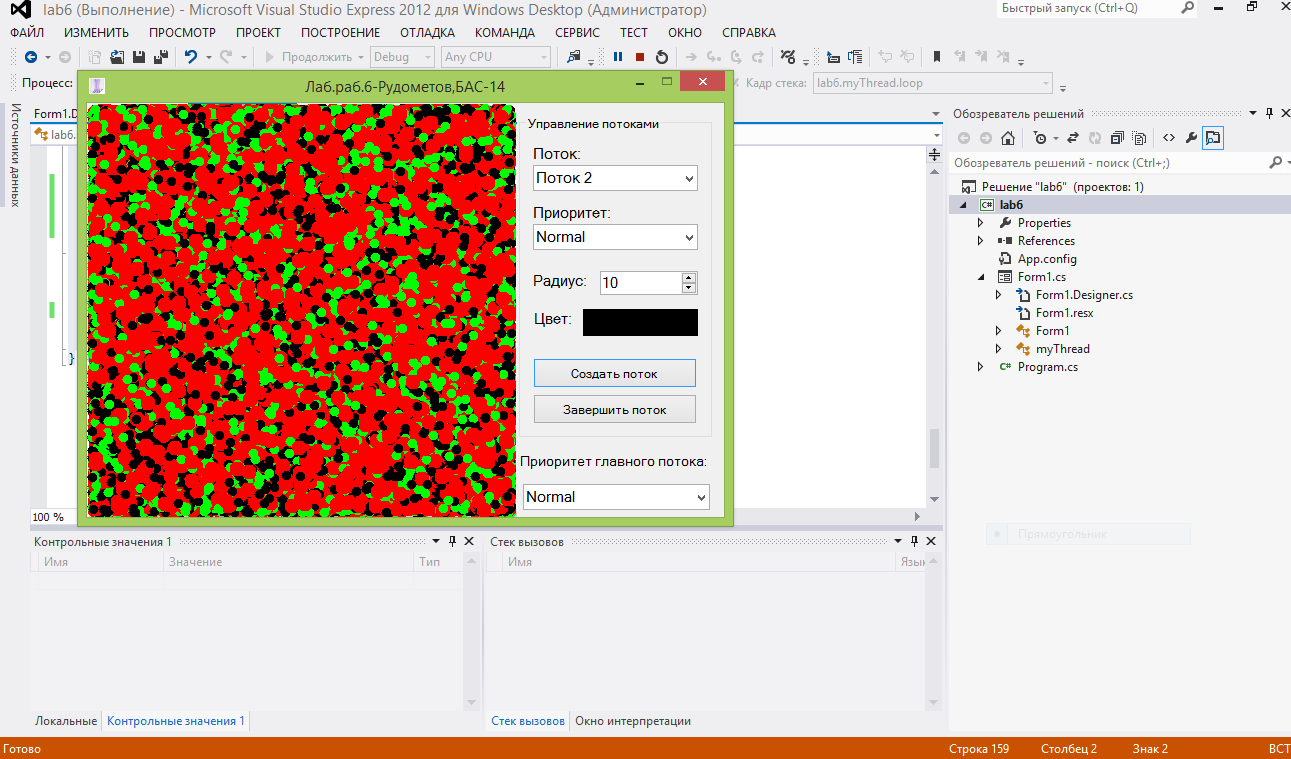
public void setSize(int s) { size = s; }

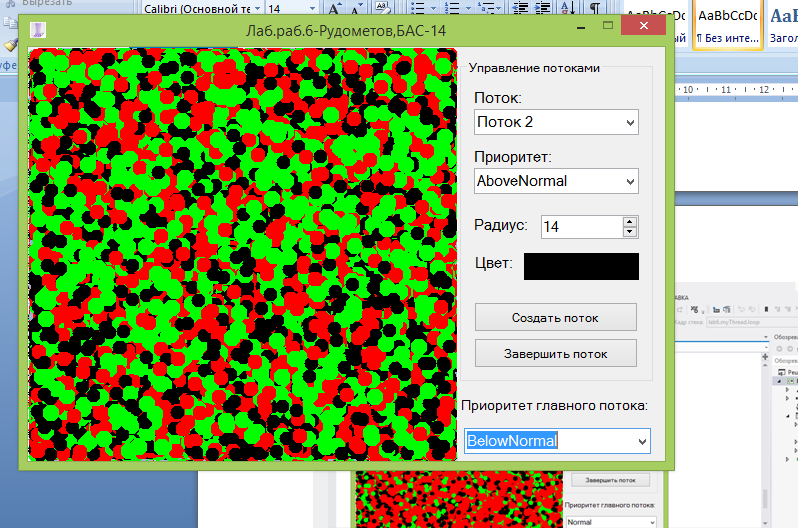
public void setPriority(ThreadPriority t) { priority = t; }

public void stop() { thread.Abort(); }

}

}





Отчёт

Лабораторная работа 6

Рудометов Глеб БАС-14

**Задание**: Разработать многопоточное приложение, позволяющие динамически управлять потоками, выводящими на экран круги произвольного радиуса и цвета.

Список «Поток:» содержит созданные в программе потоки. Изменение параметров «Приоритет:», «Радиус:» и «Цвет:» применяются к выбранному потоку. Список «Приоритет» позволяет указать приоритет потока

В поле «Радиус:» вводится радиус кругов, рисуемых текущим потоком. При щелчке на поле «Цвет:» выводится стандартное диалоговое окно для выбора цвета потока

Дополнительно на форме должны быть две командные кнопки (рис. 3): 1) при нажатии на кнопку «Создать поток» создается новый поток с параметрами по умолчанию, имя потока добавляется в список потоков; 2) нажатие кнопки «Завершить поток» уничтожает выбранный в списке поток и удаляет его из этого списка. Список «Приоритет главного потока:» позволяет задать приоритет потока формы.

**Текст программы:**

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Threading;

namespace lab6

{

public partial class Form1 : Form

{

ThreadPriority[] tp = {ThreadPriority.Lowest,ThreadPriority.BelowNormal,ThreadPriority.Normal,ThreadPriority.AboveNormal,ThreadPriority.Highest};

List<myThread> list = new List<myThread>();

int num = 0; //номер текущего потока и макс.кол-во

public Graphics g;

Color DefaultColor;

public Form1()

{

InitializeComponent();

g=panel1.CreateGraphics();

DefaultColor = pictureBox1.BackColor;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = 2; //приоритет текущего потока

comboBox1.SelectedIndex = 2; //приоритет главного потока

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

if (colorDialog1.ShowDialog() == DialogResult.OK)

{

pictureBox1.BackColor=colorDialog1.Color;

list[num].setColor(colorDialog1.Color);

}

}

private void label2\_Click(object sender, EventArgs e)

{

}

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

{

num++;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

list[num].name = "Поток " + (list.Count-1);

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

pictureBox1.BackColor = DefaultColor;

comboBox3.SelectedIndex = 2;

}

private void button2\_Click(object sender, EventArgs e)

{

list[num].stop();

if (list.Count>1)

{

comboBox2.Items.Remove(list[num].name);

list.Remove(list[num]);

if (num > 0)num--;

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void numericUpDown1\_ValueChanged(object sender, EventArgs e)

{

list[num].setSize((int)numericUpDown1.Value);

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

num = comboBox2.SelectedIndex;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

Thread.CurrentThread.Priority = tp[comboBox1.SelectedIndex];

}

private void comboBox3\_SelectedIndexChanged(object sender, EventArgs e)

{

list[num].setPriority(tp[comboBox3.SelectedIndex]);

}

}

public class myThread //класс потока

{

public int size;

public Color color;

private Thread thread;

public ThreadPriority priority;

private Graphics g;

private void loop() {

Random r=new Random((int)(new DateTime()).Ticks);

while (true) {

lock (g)

{

g.FillEllipse((Brush)new SolidBrush(color), r.Next(0, 430 - size), r.Next(0, 415 - size), size, size);

//синхронизация потоков

//Thread.Sleep(1); //чтобы дать форме с низким приоритетом хоть какой то шанс среагировать !!!!!!

}

}

} //главный цикл потока

public myThread(string name,Graphics g,int size,Color color,ThreadPriority initial) {

this.name = name;

this.size = size;

this.color = color;

this.priority = initial;

this.g = g;

thread = new Thread(new ThreadStart(loop));

thread.Start();

} //создание потока

public string name;

public void setColor(Color a) { color = a; }

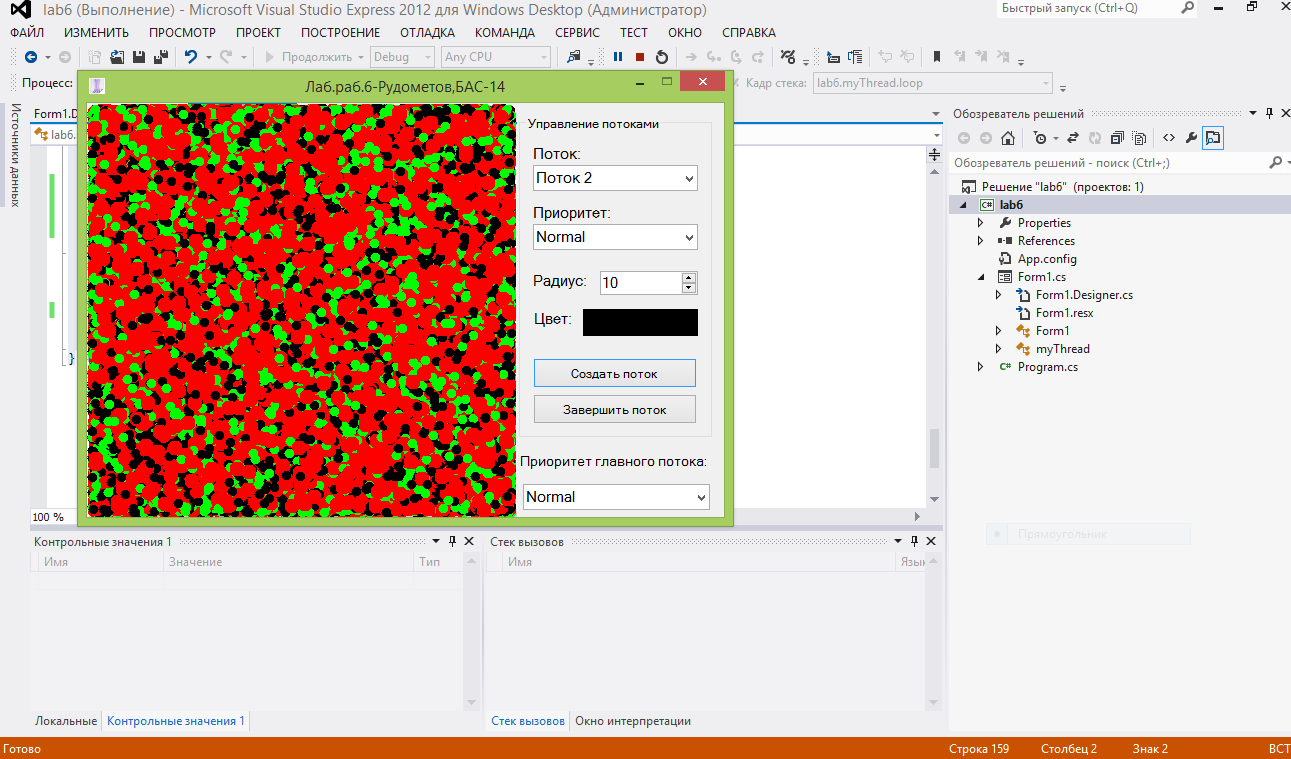
public void setSize(int s) { size = s; }

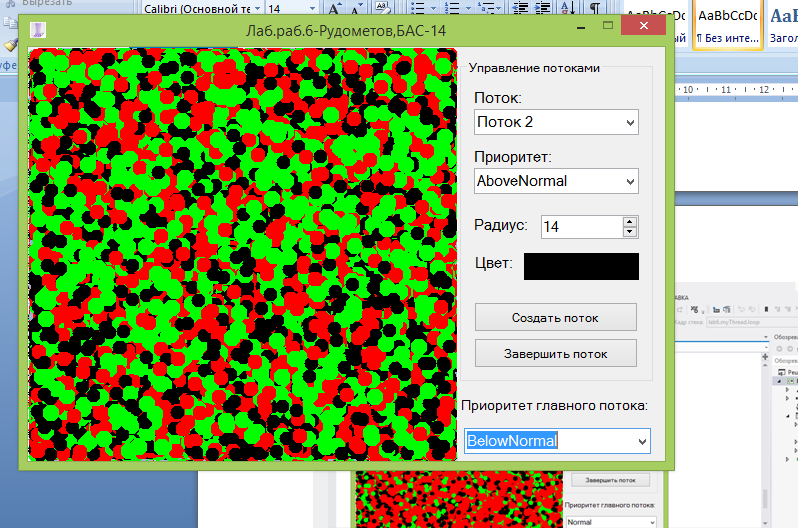
public void setPriority(ThreadPriority t) { priority = t; }

public void stop() { thread.Abort(); }

}

}





Дополнительная информация:

Отчёт

Лабораторная работа 6

Рудометов Глеб БАС-14

**Задание**: Разработать многопоточное приложение, позволяющие динамически управлять потоками, выводящими на экран круги произвольного радиуса и цвета.

Список «Поток:» содержит созданные в программе потоки. Изменение параметров «Приоритет:», «Радиус:» и «Цвет:» применяются к выбранному потоку. Список «Приоритет» позволяет указать приоритет потока

В поле «Радиус:» вводится радиус кругов, рисуемых текущим потоком. При щелчке на поле «Цвет:» выводится стандартное диалоговое окно для выбора цвета потока

Дополнительно на форме должны быть две командные кнопки (рис. 3): 1) при нажатии на кнопку «Создать поток» создается новый поток с параметрами по умолчанию, имя потока добавляется в список потоков; 2) нажатие кнопки «Завершить поток» уничтожает выбранный в списке поток и удаляет его из этого списка. Список «Приоритет главного потока:» позволяет задать приоритет потока формы.

**Текст программы:**

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Threading;

namespace lab6

{

public partial class Form1 : Form

{

ThreadPriority[] tp = {ThreadPriority.Lowest,ThreadPriority.BelowNormal,ThreadPriority.Normal,ThreadPriority.AboveNormal,ThreadPriority.Highest};

List<myThread> list = new List<myThread>();

int num = 0; //номер текущего потока и макс.кол-во

public Graphics g;

Color DefaultColor;

public Form1()

{

InitializeComponent();

g=panel1.CreateGraphics();

DefaultColor = pictureBox1.BackColor;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = 2; //приоритет текущего потока

comboBox1.SelectedIndex = 2; //приоритет главного потока

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

if (colorDialog1.ShowDialog() == DialogResult.OK)

{

pictureBox1.BackColor=colorDialog1.Color;

list[num].setColor(colorDialog1.Color);

}

}

private void label2\_Click(object sender, EventArgs e)

{

}

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

{

num++;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

list[num].name = "Поток " + (list.Count-1);

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

pictureBox1.BackColor = DefaultColor;

comboBox3.SelectedIndex = 2;

}

private void button2\_Click(object sender, EventArgs e)

{

list[num].stop();

if (list.Count>1)

{

comboBox2.Items.Remove(list[num].name);

list.Remove(list[num]);

if (num > 0)num--;

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void numericUpDown1\_ValueChanged(object sender, EventArgs e)

{

list[num].setSize((int)numericUpDown1.Value);

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

num = comboBox2.SelectedIndex;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

Thread.CurrentThread.Priority = tp[comboBox1.SelectedIndex];

}

private void comboBox3\_SelectedIndexChanged(object sender, EventArgs e)

{

list[num].setPriority(tp[comboBox3.SelectedIndex]);

}

}

public class myThread //класс потока

{

public int size;

public Color color;

private Thread thread;

public ThreadPriority priority;

private Graphics g;

private void loop() {

Random r=new Random((int)(new DateTime()).Ticks);

while (true) {

lock (g)

{

g.FillEllipse((Brush)new SolidBrush(color), r.Next(0, 430 - size), r.Next(0, 415 - size), size, size);

//синхронизация потоков

//Thread.Sleep(1); //чтобы дать форме с низким приоритетом хоть какой то шанс среагировать !!!!!!

}

}

} //главный цикл потока

public myThread(string name,Graphics g,int size,Color color,ThreadPriority initial) {

this.name = name;

this.size = size;

this.color = color;

this.priority = initial;

this.g = g;

thread = new Thread(new ThreadStart(loop));

thread.Start();

} //создание потока

public string name;

public void setColor(Color a) { color = a; }

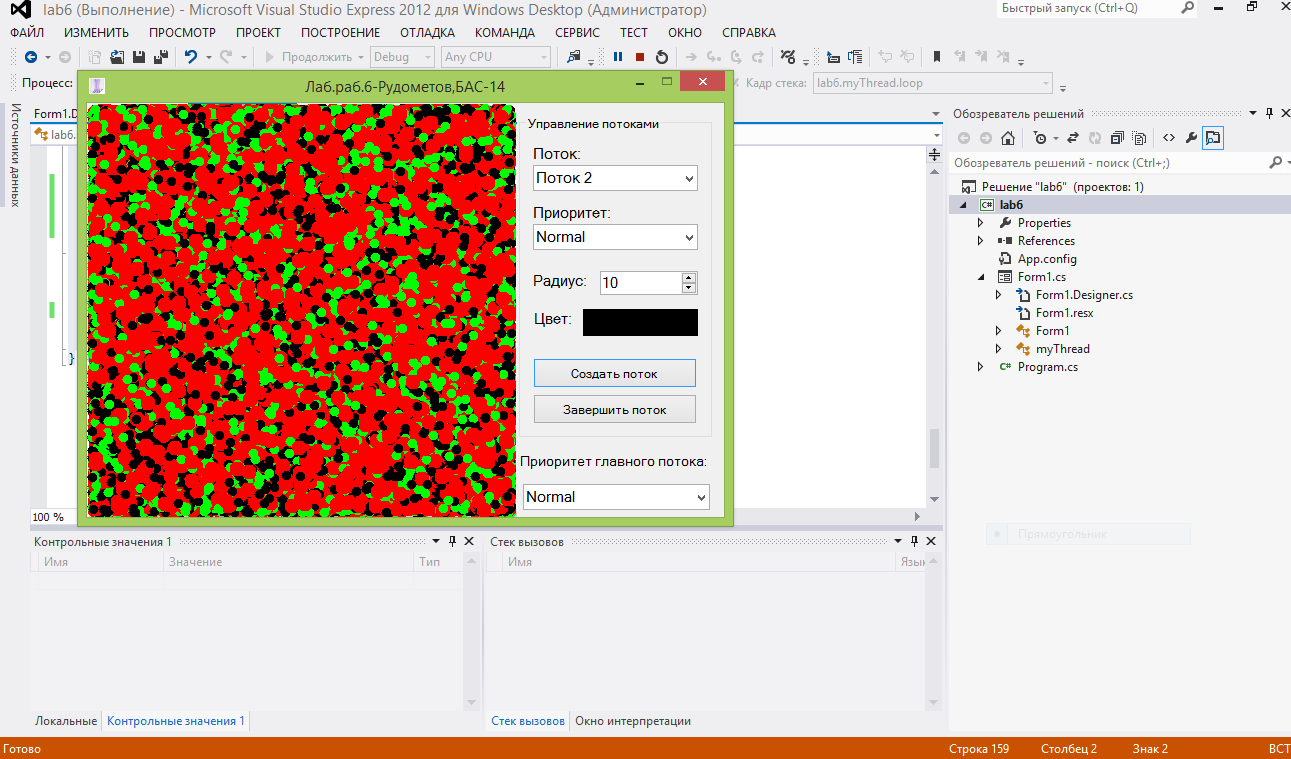
public void setSize(int s) { size = s; }

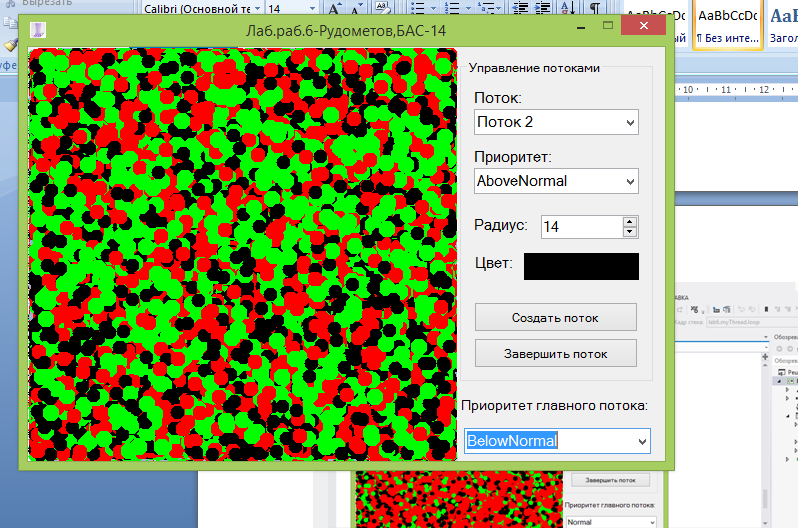
public void setPriority(ThreadPriority t) { priority = t; }

public void stop() { thread.Abort(); }

}

}





Отчёт

Лабораторная работа 6

Рудометов Глеб БАС-14

**Задание**: Разработать многопоточное приложение, позволяющие динамически управлять потоками, выводящими на экран круги произвольного радиуса и цвета.

Список «Поток:» содержит созданные в программе потоки. Изменение параметров «Приоритет:», «Радиус:» и «Цвет:» применяются к выбранному потоку. Список «Приоритет» позволяет указать приоритет потока

В поле «Радиус:» вводится радиус кругов, рисуемых текущим потоком. При щелчке на поле «Цвет:» выводится стандартное диалоговое окно для выбора цвета потока

Дополнительно на форме должны быть две командные кнопки (рис. 3): 1) при нажатии на кнопку «Создать поток» создается новый поток с параметрами по умолчанию, имя потока добавляется в список потоков; 2) нажатие кнопки «Завершить поток» уничтожает выбранный в списке поток и удаляет его из этого списка. Список «Приоритет главного потока:» позволяет задать приоритет потока формы.

**Текст программы:**

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Threading;

namespace lab6

{

public partial class Form1 : Form

{

ThreadPriority[] tp = {ThreadPriority.Lowest,ThreadPriority.BelowNormal,ThreadPriority.Normal,ThreadPriority.AboveNormal,ThreadPriority.Highest};

List<myThread> list = new List<myThread>();

int num = 0; //номер текущего потока и макс.кол-во

public Graphics g;

Color DefaultColor;

public Form1()

{

InitializeComponent();

g=panel1.CreateGraphics();

DefaultColor = pictureBox1.BackColor;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = 2; //приоритет текущего потока

comboBox1.SelectedIndex = 2; //приоритет главного потока

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

if (colorDialog1.ShowDialog() == DialogResult.OK)

{

pictureBox1.BackColor=colorDialog1.Color;

list[num].setColor(colorDialog1.Color);

}

}

private void label2\_Click(object sender, EventArgs e)

{

}

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

{

num++;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

list[num].name = "Поток " + (list.Count-1);

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

pictureBox1.BackColor = DefaultColor;

comboBox3.SelectedIndex = 2;

}

private void button2\_Click(object sender, EventArgs e)

{

list[num].stop();

if (list.Count>1)

{

comboBox2.Items.Remove(list[num].name);

list.Remove(list[num]);

if (num > 0)num--;

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void numericUpDown1\_ValueChanged(object sender, EventArgs e)

{

list[num].setSize((int)numericUpDown1.Value);

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

num = comboBox2.SelectedIndex;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

Thread.CurrentThread.Priority = tp[comboBox1.SelectedIndex];

}

private void comboBox3\_SelectedIndexChanged(object sender, EventArgs e)

{

list[num].setPriority(tp[comboBox3.SelectedIndex]);

}

}

public class myThread //класс потока

{

public int size;

public Color color;

private Thread thread;

public ThreadPriority priority;

private Graphics g;

private void loop() {

Random r=new Random((int)(new DateTime()).Ticks);

while (true) {

lock (g)

{

g.FillEllipse((Brush)new SolidBrush(color), r.Next(0, 430 - size), r.Next(0, 415 - size), size, size);

//синхронизация потоков

//Thread.Sleep(1); //чтобы дать форме с низким приоритетом хоть какой то шанс среагировать !!!!!!

}

}

} //главный цикл потока

public myThread(string name,Graphics g,int size,Color color,ThreadPriority initial) {

this.name = name;

this.size = size;

this.color = color;

this.priority = initial;

this.g = g;

thread = new Thread(new ThreadStart(loop));

thread.Start();

} //создание потока

public string name;

public void setColor(Color a) { color = a; }

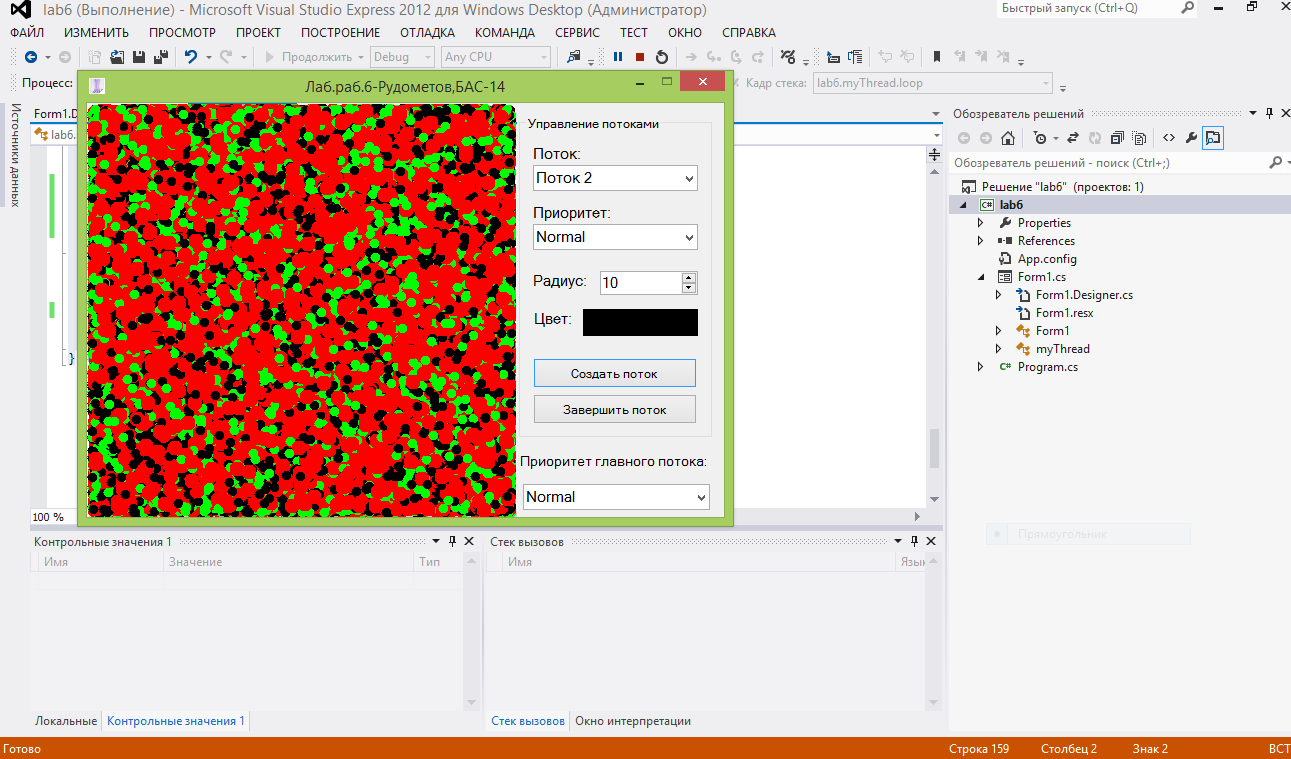
public void setSize(int s) { size = s; }

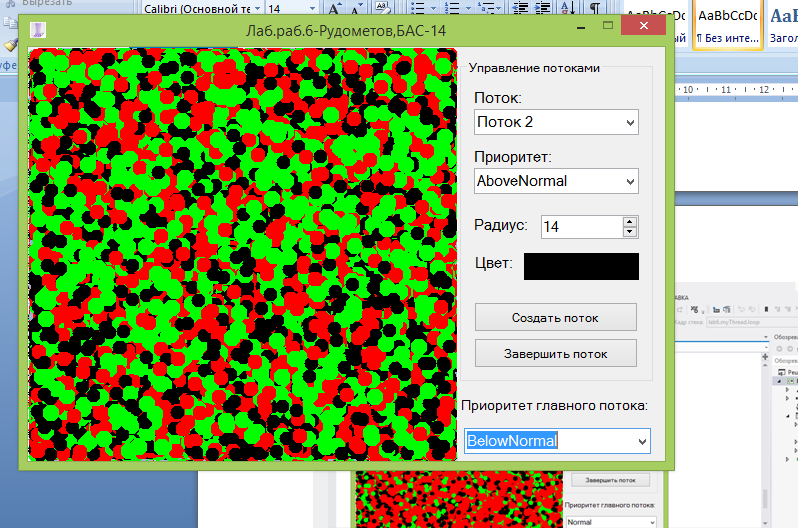
public void setPriority(ThreadPriority t) { priority = t; }

public void stop() { thread.Abort(); }

}

}





Дополнительная информация:

Отчёт

Лабораторная работа 6

Рудометов Глеб БАС-14

**Задание**: Разработать многопоточное приложение, позволяющие динамически управлять потоками, выводящими на экран круги произвольного радиуса и цвета.

Список «Поток:» содержит созданные в программе потоки. Изменение параметров «Приоритет:», «Радиус:» и «Цвет:» применяются к выбранному потоку. Список «Приоритет» позволяет указать приоритет потока

В поле «Радиус:» вводится радиус кругов, рисуемых текущим потоком. При щелчке на поле «Цвет:» выводится стандартное диалоговое окно для выбора цвета потока

Дополнительно на форме должны быть две командные кнопки (рис. 3): 1) при нажатии на кнопку «Создать поток» создается новый поток с параметрами по умолчанию, имя потока добавляется в список потоков; 2) нажатие кнопки «Завершить поток» уничтожает выбранный в списке поток и удаляет его из этого списка. Список «Приоритет главного потока:» позволяет задать приоритет потока формы.

**Текст программы:**

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Threading;

namespace lab6

{

public partial class Form1 : Form

{

ThreadPriority[] tp = {ThreadPriority.Lowest,ThreadPriority.BelowNormal,ThreadPriority.Normal,ThreadPriority.AboveNormal,ThreadPriority.Highest};

List<myThread> list = new List<myThread>();

int num = 0; //номер текущего потока и макс.кол-во

public Graphics g;

Color DefaultColor;

public Form1()

{

InitializeComponent();

g=panel1.CreateGraphics();

DefaultColor = pictureBox1.BackColor;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = 2; //приоритет текущего потока

comboBox1.SelectedIndex = 2; //приоритет главного потока

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

if (colorDialog1.ShowDialog() == DialogResult.OK)

{

pictureBox1.BackColor=colorDialog1.Color;

list[num].setColor(colorDialog1.Color);

}

}

private void label2\_Click(object sender, EventArgs e)

{

}

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

{

num++;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

list[num].name = "Поток " + (list.Count-1);

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

pictureBox1.BackColor = DefaultColor;

comboBox3.SelectedIndex = 2;

}

private void button2\_Click(object sender, EventArgs e)

{

list[num].stop();

if (list.Count>1)

{

comboBox2.Items.Remove(list[num].name);

list.Remove(list[num]);

if (num > 0)num--;

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void numericUpDown1\_ValueChanged(object sender, EventArgs e)

{

list[num].setSize((int)numericUpDown1.Value);

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

num = comboBox2.SelectedIndex;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

Thread.CurrentThread.Priority = tp[comboBox1.SelectedIndex];

}

private void comboBox3\_SelectedIndexChanged(object sender, EventArgs e)

{

list[num].setPriority(tp[comboBox3.SelectedIndex]);

}

}

public class myThread //класс потока

{

public int size;

public Color color;

private Thread thread;

public ThreadPriority priority;

private Graphics g;

private void loop() {

Random r=new Random((int)(new DateTime()).Ticks);

while (true) {

lock (g)

{

g.FillEllipse((Brush)new SolidBrush(color), r.Next(0, 430 - size), r.Next(0, 415 - size), size, size);

//синхронизация потоков

//Thread.Sleep(1); //чтобы дать форме с низким приоритетом хоть какой то шанс среагировать !!!!!!

}

}

} //главный цикл потока

public myThread(string name,Graphics g,int size,Color color,ThreadPriority initial) {

this.name = name;

this.size = size;

this.color = color;

this.priority = initial;

this.g = g;

thread = new Thread(new ThreadStart(loop));

thread.Start();

} //создание потока

public string name;

public void setColor(Color a) { color = a; }

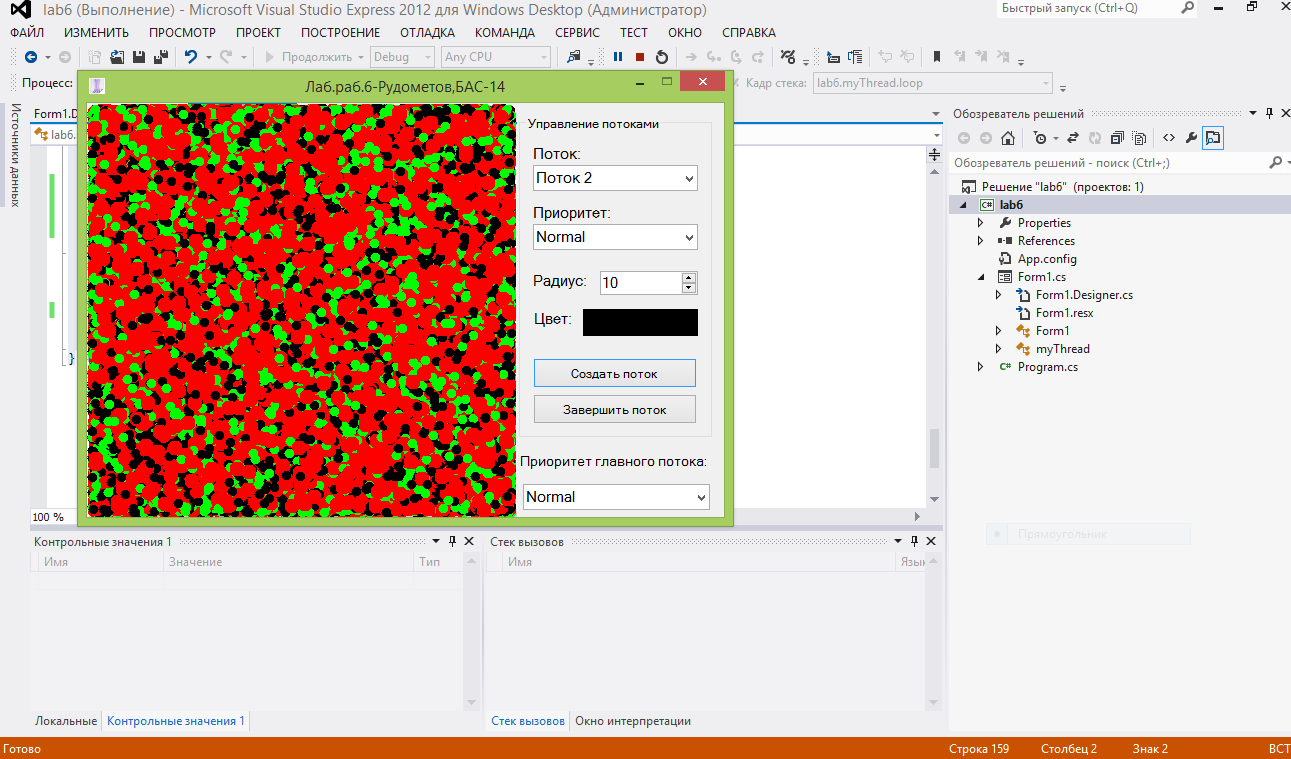
public void setSize(int s) { size = s; }

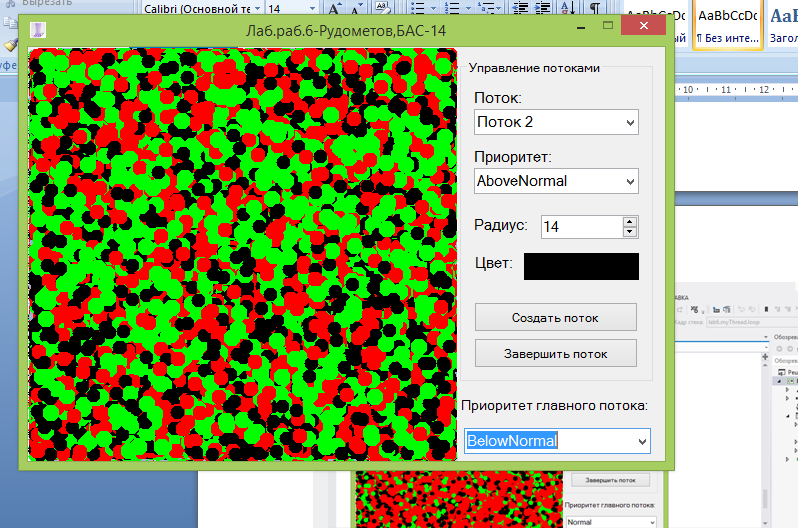
public void setPriority(ThreadPriority t) { priority = t; }

public void stop() { thread.Abort(); }

}

}





Отчёт

Лабораторная работа 6

Рудометов Глеб БАС-14

**Задание**: Разработать многопоточное приложение, позволяющие динамически управлять потоками, выводящими на экран круги произвольного радиуса и цвета.

Список «Поток:» содержит созданные в программе потоки. Изменение параметров «Приоритет:», «Радиус:» и «Цвет:» применяются к выбранному потоку. Список «Приоритет» позволяет указать приоритет потока

В поле «Радиус:» вводится радиус кругов, рисуемых текущим потоком. При щелчке на поле «Цвет:» выводится стандартное диалоговое окно для выбора цвета потока

Дополнительно на форме должны быть две командные кнопки (рис. 3): 1) при нажатии на кнопку «Создать поток» создается новый поток с параметрами по умолчанию, имя потока добавляется в список потоков; 2) нажатие кнопки «Завершить поток» уничтожает выбранный в списке поток и удаляет его из этого списка. Список «Приоритет главного потока:» позволяет задать приоритет потока формы.

**Текст программы:**

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Threading;

namespace lab6

{

public partial class Form1 : Form

{

ThreadPriority[] tp = {ThreadPriority.Lowest,ThreadPriority.BelowNormal,ThreadPriority.Normal,ThreadPriority.AboveNormal,ThreadPriority.Highest};

List<myThread> list = new List<myThread>();

int num = 0; //номер текущего потока и макс.кол-во

public Graphics g;

Color DefaultColor;

public Form1()

{

InitializeComponent();

g=panel1.CreateGraphics();

DefaultColor = pictureBox1.BackColor;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = 2; //приоритет текущего потока

comboBox1.SelectedIndex = 2; //приоритет главного потока

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

if (colorDialog1.ShowDialog() == DialogResult.OK)

{

pictureBox1.BackColor=colorDialog1.Color;

list[num].setColor(colorDialog1.Color);

}

}

private void label2\_Click(object sender, EventArgs e)

{

}

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

{

num++;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

list[num].name = "Поток " + (list.Count-1);

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

pictureBox1.BackColor = DefaultColor;

comboBox3.SelectedIndex = 2;

}

private void button2\_Click(object sender, EventArgs e)

{

list[num].stop();

if (list.Count>1)

{

comboBox2.Items.Remove(list[num].name);

list.Remove(list[num]);

if (num > 0)num--;

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void numericUpDown1\_ValueChanged(object sender, EventArgs e)

{

list[num].setSize((int)numericUpDown1.Value);

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

num = comboBox2.SelectedIndex;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

Thread.CurrentThread.Priority = tp[comboBox1.SelectedIndex];

}

private void comboBox3\_SelectedIndexChanged(object sender, EventArgs e)

{

list[num].setPriority(tp[comboBox3.SelectedIndex]);

}

}

public class myThread //класс потока

{

public int size;

public Color color;

private Thread thread;

public ThreadPriority priority;

private Graphics g;

private void loop() {

Random r=new Random((int)(new DateTime()).Ticks);

while (true) {

lock (g)

{

g.FillEllipse((Brush)new SolidBrush(color), r.Next(0, 430 - size), r.Next(0, 415 - size), size, size);

//синхронизация потоков

//Thread.Sleep(1); //чтобы дать форме с низким приоритетом хоть какой то шанс среагировать !!!!!!

}

}

} //главный цикл потока

public myThread(string name,Graphics g,int size,Color color,ThreadPriority initial) {

this.name = name;

this.size = size;

this.color = color;

this.priority = initial;

this.g = g;

thread = new Thread(new ThreadStart(loop));

thread.Start();

} //создание потока

public string name;

public void setColor(Color a) { color = a; }

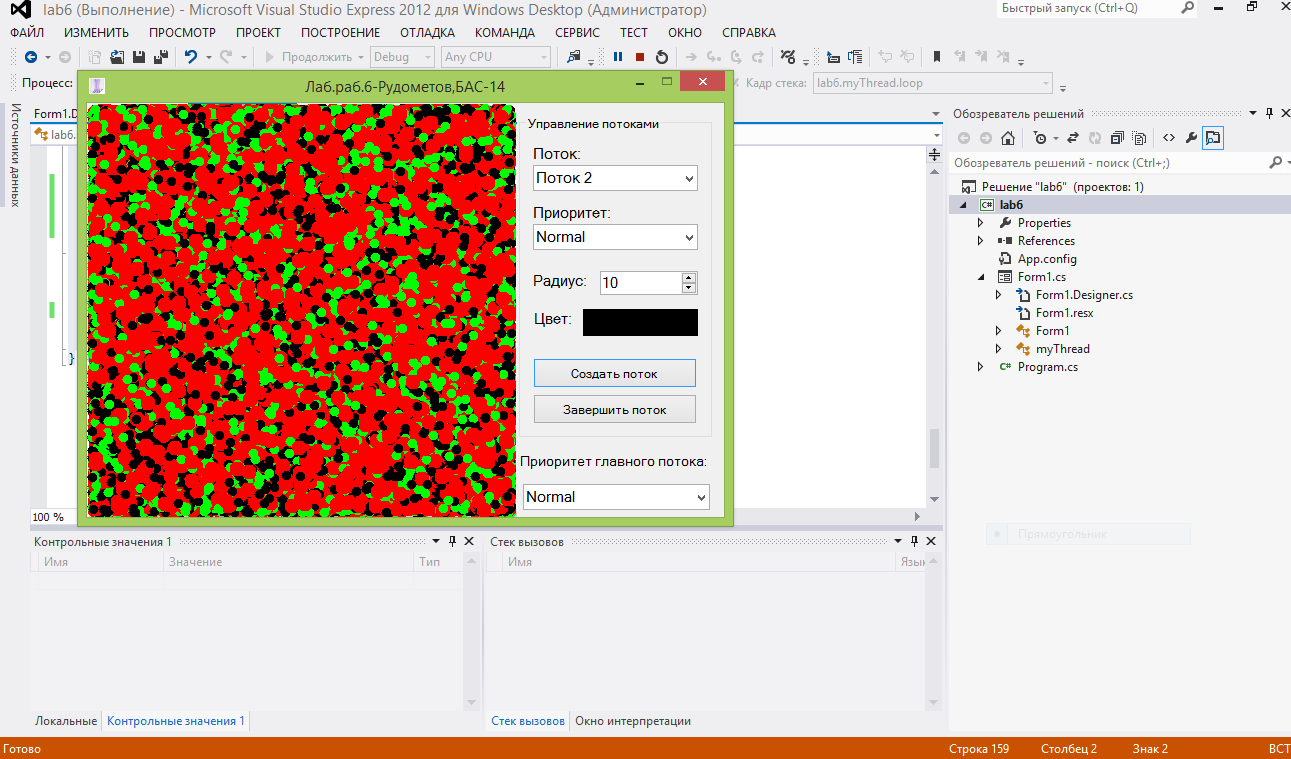
public void setSize(int s) { size = s; }

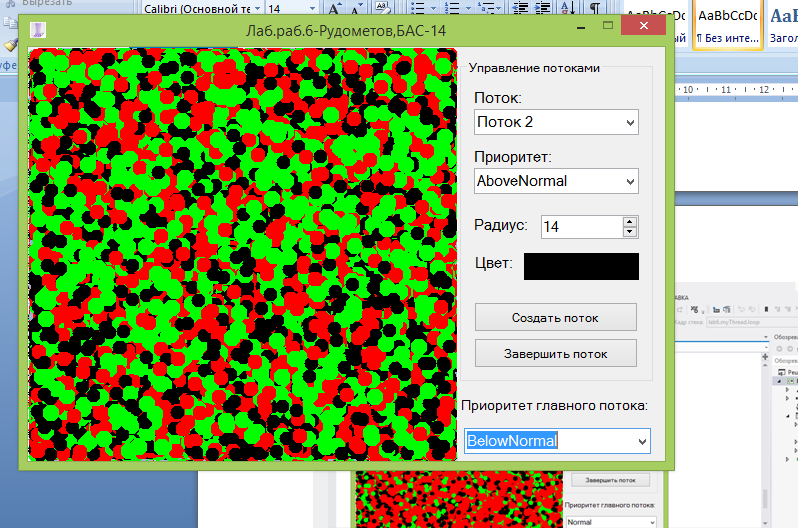
public void setPriority(ThreadPriority t) { priority = t; }

public void stop() { thread.Abort(); }

}

}





Дополнительная информация:

Отчёт

Лабораторная работа 6

Рудометов Глеб БАС-14

**Задание**: Разработать многопоточное приложение, позволяющие динамически управлять потоками, выводящими на экран круги произвольного радиуса и цвета.

Список «Поток:» содержит созданные в программе потоки. Изменение параметров «Приоритет:», «Радиус:» и «Цвет:» применяются к выбранному потоку. Список «Приоритет» позволяет указать приоритет потока

В поле «Радиус:» вводится радиус кругов, рисуемых текущим потоком. При щелчке на поле «Цвет:» выводится стандартное диалоговое окно для выбора цвета потока

Дополнительно на форме должны быть две командные кнопки (рис. 3): 1) при нажатии на кнопку «Создать поток» создается новый поток с параметрами по умолчанию, имя потока добавляется в список потоков; 2) нажатие кнопки «Завершить поток» уничтожает выбранный в списке поток и удаляет его из этого списка. Список «Приоритет главного потока:» позволяет задать приоритет потока формы.

**Текст программы:**

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Threading;

namespace lab6

{

public partial class Form1 : Form

{

ThreadPriority[] tp = {ThreadPriority.Lowest,ThreadPriority.BelowNormal,ThreadPriority.Normal,ThreadPriority.AboveNormal,ThreadPriority.Highest};

List<myThread> list = new List<myThread>();

int num = 0; //номер текущего потока и макс.кол-во

public Graphics g;

Color DefaultColor;

public Form1()

{

InitializeComponent();

g=panel1.CreateGraphics();

DefaultColor = pictureBox1.BackColor;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = 2; //приоритет текущего потока

comboBox1.SelectedIndex = 2; //приоритет главного потока

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

if (colorDialog1.ShowDialog() == DialogResult.OK)

{

pictureBox1.BackColor=colorDialog1.Color;

list[num].setColor(colorDialog1.Color);

}

}

private void label2\_Click(object sender, EventArgs e)

{

}

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

{

num++;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

list[num].name = "Поток " + (list.Count-1);

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

pictureBox1.BackColor = DefaultColor;

comboBox3.SelectedIndex = 2;

}

private void button2\_Click(object sender, EventArgs e)

{

list[num].stop();

if (list.Count>1)

{

comboBox2.Items.Remove(list[num].name);

list.Remove(list[num]);

if (num > 0)num--;

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void numericUpDown1\_ValueChanged(object sender, EventArgs e)

{

list[num].setSize((int)numericUpDown1.Value);

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

num = comboBox2.SelectedIndex;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

Thread.CurrentThread.Priority = tp[comboBox1.SelectedIndex];

}

private void comboBox3\_SelectedIndexChanged(object sender, EventArgs e)

{

list[num].setPriority(tp[comboBox3.SelectedIndex]);

}

}

public class myThread //класс потока

{

public int size;

public Color color;

private Thread thread;

public ThreadPriority priority;

private Graphics g;

private void loop() {

Random r=new Random((int)(new DateTime()).Ticks);

while (true) {

lock (g)

{

g.FillEllipse((Brush)new SolidBrush(color), r.Next(0, 430 - size), r.Next(0, 415 - size), size, size);

//синхронизация потоков

//Thread.Sleep(1); //чтобы дать форме с низким приоритетом хоть какой то шанс среагировать !!!!!!

}

}

} //главный цикл потока

public myThread(string name,Graphics g,int size,Color color,ThreadPriority initial) {

this.name = name;

this.size = size;

this.color = color;

this.priority = initial;

this.g = g;

thread = new Thread(new ThreadStart(loop));

thread.Start();

} //создание потока

public string name;

public void setColor(Color a) { color = a; }

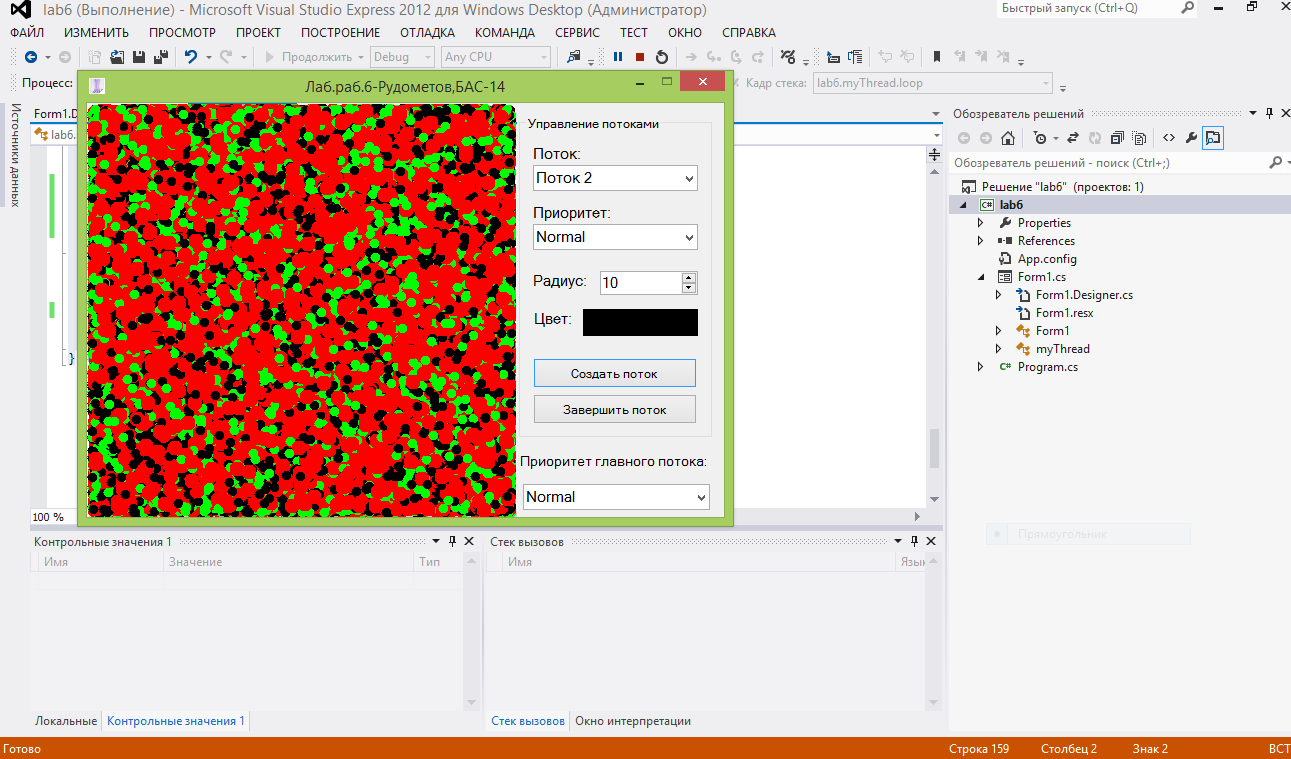
public void setSize(int s) { size = s; }

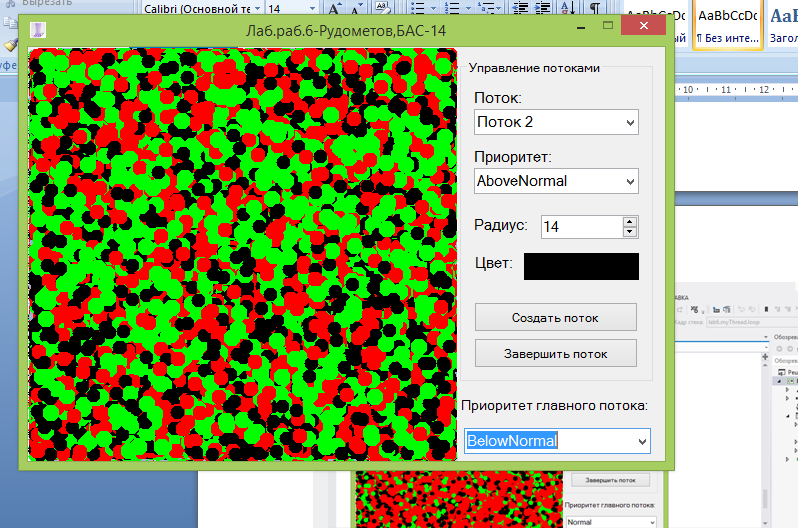
public void setPriority(ThreadPriority t) { priority = t; }

public void stop() { thread.Abort(); }

}

}





Отчёт

Лабораторная работа 6

Рудометов Глеб БАС-14

**Задание**: Разработать многопоточное приложение, позволяющие динамически управлять потоками, выводящими на экран круги произвольного радиуса и цвета.

Список «Поток:» содержит созданные в программе потоки. Изменение параметров «Приоритет:», «Радиус:» и «Цвет:» применяются к выбранному потоку. Список «Приоритет» позволяет указать приоритет потока

В поле «Радиус:» вводится радиус кругов, рисуемых текущим потоком. При щелчке на поле «Цвет:» выводится стандартное диалоговое окно для выбора цвета потока

Дополнительно на форме должны быть две командные кнопки (рис. 3): 1) при нажатии на кнопку «Создать поток» создается новый поток с параметрами по умолчанию, имя потока добавляется в список потоков; 2) нажатие кнопки «Завершить поток» уничтожает выбранный в списке поток и удаляет его из этого списка. Список «Приоритет главного потока:» позволяет задать приоритет потока формы.

**Текст программы:**

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Threading;

namespace lab6

{

public partial class Form1 : Form

{

ThreadPriority[] tp = {ThreadPriority.Lowest,ThreadPriority.BelowNormal,ThreadPriority.Normal,ThreadPriority.AboveNormal,ThreadPriority.Highest};

List<myThread> list = new List<myThread>();

int num = 0; //номер текущего потока и макс.кол-во

public Graphics g;

Color DefaultColor;

public Form1()

{

InitializeComponent();

g=panel1.CreateGraphics();

DefaultColor = pictureBox1.BackColor;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = 2; //приоритет текущего потока

comboBox1.SelectedIndex = 2; //приоритет главного потока

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

if (colorDialog1.ShowDialog() == DialogResult.OK)

{

pictureBox1.BackColor=colorDialog1.Color;

list[num].setColor(colorDialog1.Color);

}

}

private void label2\_Click(object sender, EventArgs e)

{

}

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

{

num++;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

list[num].name = "Поток " + (list.Count-1);

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

pictureBox1.BackColor = DefaultColor;

comboBox3.SelectedIndex = 2;

}

private void button2\_Click(object sender, EventArgs e)

{

list[num].stop();

if (list.Count>1)

{

comboBox2.Items.Remove(list[num].name);

list.Remove(list[num]);

if (num > 0)num--;

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void numericUpDown1\_ValueChanged(object sender, EventArgs e)

{

list[num].setSize((int)numericUpDown1.Value);

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

num = comboBox2.SelectedIndex;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

Thread.CurrentThread.Priority = tp[comboBox1.SelectedIndex];

}

private void comboBox3\_SelectedIndexChanged(object sender, EventArgs e)

{

list[num].setPriority(tp[comboBox3.SelectedIndex]);

}

}

public class myThread //класс потока

{

public int size;

public Color color;

private Thread thread;

public ThreadPriority priority;

private Graphics g;

private void loop() {

Random r=new Random((int)(new DateTime()).Ticks);

while (true) {

lock (g)

{

g.FillEllipse((Brush)new SolidBrush(color), r.Next(0, 430 - size), r.Next(0, 415 - size), size, size);

//синхронизация потоков

//Thread.Sleep(1); //чтобы дать форме с низким приоритетом хоть какой то шанс среагировать !!!!!!

}

}

} //главный цикл потока

public myThread(string name,Graphics g,int size,Color color,ThreadPriority initial) {

this.name = name;

this.size = size;

this.color = color;

this.priority = initial;

this.g = g;

thread = new Thread(new ThreadStart(loop));

thread.Start();

} //создание потока

public string name;

public void setColor(Color a) { color = a; }

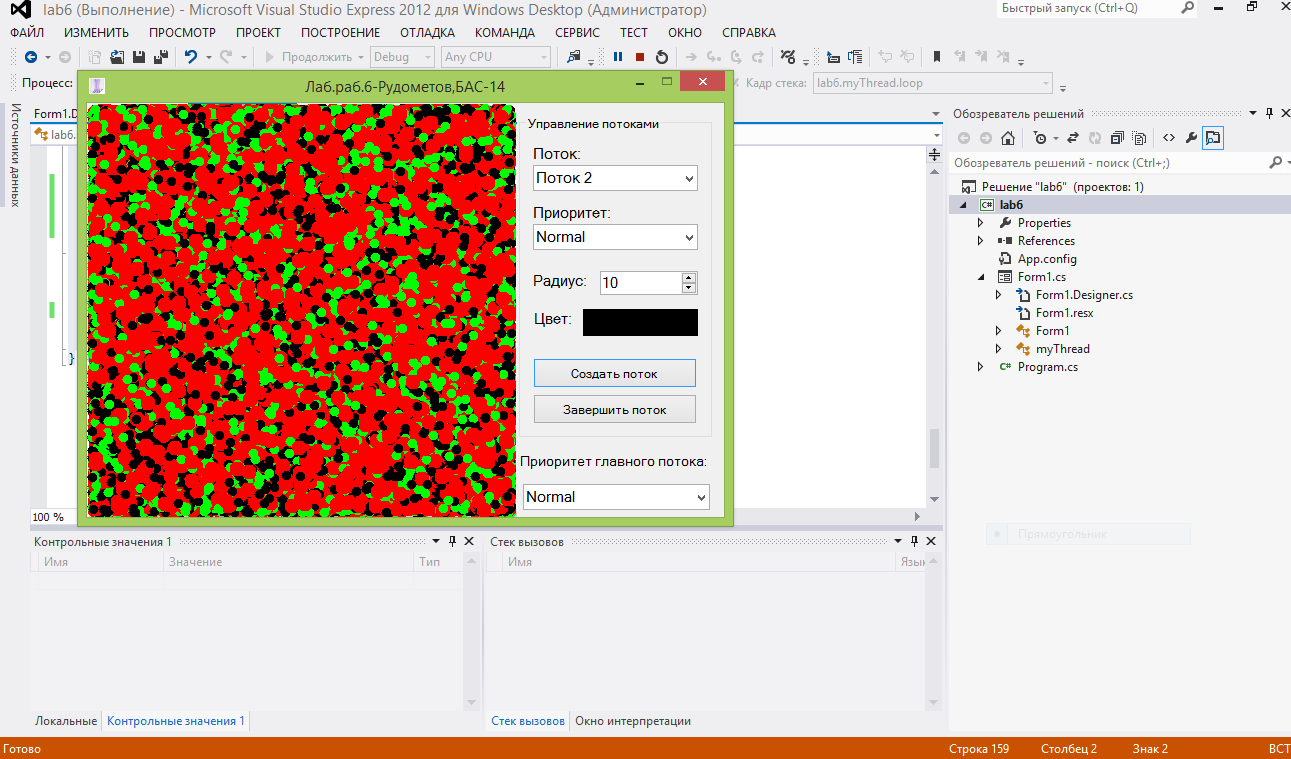
public void setSize(int s) { size = s; }

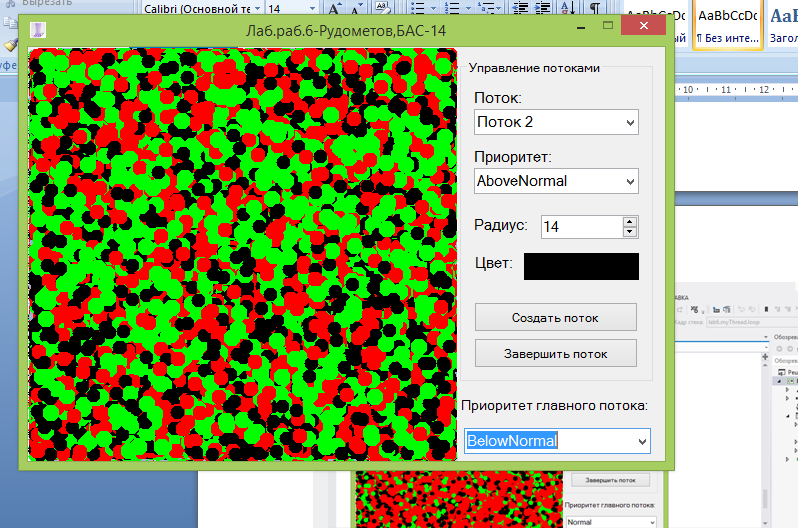
public void setPriority(ThreadPriority t) { priority = t; }

public void stop() { thread.Abort(); }

}

}





Дополнительная информация:

Отчёт

Лабораторная работа 6

Рудометов Глеб БАС-14

**Задание**: Разработать многопоточное приложение, позволяющие динамически управлять потоками, выводящими на экран круги произвольного радиуса и цвета.

Список «Поток:» содержит созданные в программе потоки. Изменение параметров «Приоритет:», «Радиус:» и «Цвет:» применяются к выбранному потоку. Список «Приоритет» позволяет указать приоритет потока

В поле «Радиус:» вводится радиус кругов, рисуемых текущим потоком. При щелчке на поле «Цвет:» выводится стандартное диалоговое окно для выбора цвета потока

Дополнительно на форме должны быть две командные кнопки (рис. 3): 1) при нажатии на кнопку «Создать поток» создается новый поток с параметрами по умолчанию, имя потока добавляется в список потоков; 2) нажатие кнопки «Завершить поток» уничтожает выбранный в списке поток и удаляет его из этого списка. Список «Приоритет главного потока:» позволяет задать приоритет потока формы.

**Текст программы:**

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Threading;

namespace lab6

{

public partial class Form1 : Form

{

ThreadPriority[] tp = {ThreadPriority.Lowest,ThreadPriority.BelowNormal,ThreadPriority.Normal,ThreadPriority.AboveNormal,ThreadPriority.Highest};

List<myThread> list = new List<myThread>();

int num = 0; //номер текущего потока и макс.кол-во

public Graphics g;

Color DefaultColor;

public Form1()

{

InitializeComponent();

g=panel1.CreateGraphics();

DefaultColor = pictureBox1.BackColor;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = 2; //приоритет текущего потока

comboBox1.SelectedIndex = 2; //приоритет главного потока

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

if (colorDialog1.ShowDialog() == DialogResult.OK)

{

pictureBox1.BackColor=colorDialog1.Color;

list[num].setColor(colorDialog1.Color);

}

}

private void label2\_Click(object sender, EventArgs e)

{

}

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

{

num++;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

list[num].name = "Поток " + (list.Count-1);

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

pictureBox1.BackColor = DefaultColor;

comboBox3.SelectedIndex = 2;

}

private void button2\_Click(object sender, EventArgs e)

{

list[num].stop();

if (list.Count>1)

{

comboBox2.Items.Remove(list[num].name);

list.Remove(list[num]);

if (num > 0)num--;

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void numericUpDown1\_ValueChanged(object sender, EventArgs e)

{

list[num].setSize((int)numericUpDown1.Value);

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

num = comboBox2.SelectedIndex;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

Thread.CurrentThread.Priority = tp[comboBox1.SelectedIndex];

}

private void comboBox3\_SelectedIndexChanged(object sender, EventArgs e)

{

list[num].setPriority(tp[comboBox3.SelectedIndex]);

}

}

public class myThread //класс потока

{

public int size;

public Color color;

private Thread thread;

public ThreadPriority priority;

private Graphics g;

private void loop() {

Random r=new Random((int)(new DateTime()).Ticks);

while (true) {

lock (g)

{

g.FillEllipse((Brush)new SolidBrush(color), r.Next(0, 430 - size), r.Next(0, 415 - size), size, size);

//синхронизация потоков

//Thread.Sleep(1); //чтобы дать форме с низким приоритетом хоть какой то шанс среагировать !!!!!!

}

}

} //главный цикл потока

public myThread(string name,Graphics g,int size,Color color,ThreadPriority initial) {

this.name = name;

this.size = size;

this.color = color;

this.priority = initial;

this.g = g;

thread = new Thread(new ThreadStart(loop));

thread.Start();

} //создание потока

public string name;

public void setColor(Color a) { color = a; }

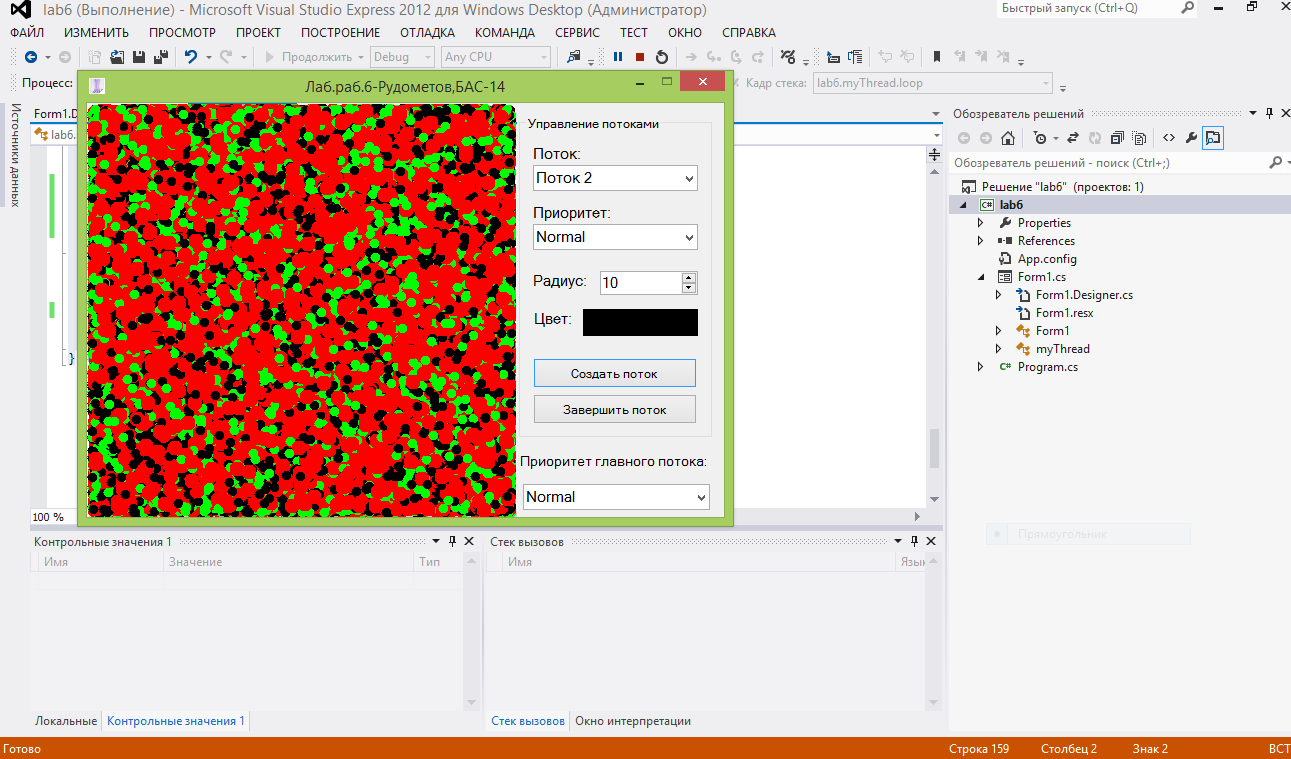
public void setSize(int s) { size = s; }

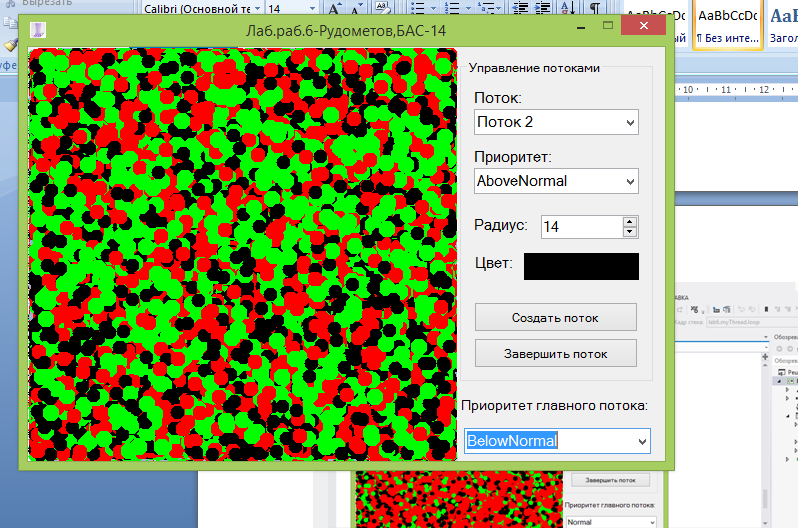
public void setPriority(ThreadPriority t) { priority = t; }

public void stop() { thread.Abort(); }

}

}





Отчёт

Лабораторная работа 6

Рудометов Глеб БАС-14

**Задание**: Разработать многопоточное приложение, позволяющие динамически управлять потоками, выводящими на экран круги произвольного радиуса и цвета.

Список «Поток:» содержит созданные в программе потоки. Изменение параметров «Приоритет:», «Радиус:» и «Цвет:» применяются к выбранному потоку. Список «Приоритет» позволяет указать приоритет потока

В поле «Радиус:» вводится радиус кругов, рисуемых текущим потоком. При щелчке на поле «Цвет:» выводится стандартное диалоговое окно для выбора цвета потока

Дополнительно на форме должны быть две командные кнопки (рис. 3): 1) при нажатии на кнопку «Создать поток» создается новый поток с параметрами по умолчанию, имя потока добавляется в список потоков; 2) нажатие кнопки «Завершить поток» уничтожает выбранный в списке поток и удаляет его из этого списка. Список «Приоритет главного потока:» позволяет задать приоритет потока формы.

**Текст программы:**

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Threading;

namespace lab6

{

public partial class Form1 : Form

{

ThreadPriority[] tp = {ThreadPriority.Lowest,ThreadPriority.BelowNormal,ThreadPriority.Normal,ThreadPriority.AboveNormal,ThreadPriority.Highest};

List<myThread> list = new List<myThread>();

int num = 0; //номер текущего потока и макс.кол-во

public Graphics g;

Color DefaultColor;

public Form1()

{

InitializeComponent();

g=panel1.CreateGraphics();

DefaultColor = pictureBox1.BackColor;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = 2; //приоритет текущего потока

comboBox1.SelectedIndex = 2; //приоритет главного потока

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

if (colorDialog1.ShowDialog() == DialogResult.OK)

{

pictureBox1.BackColor=colorDialog1.Color;

list[num].setColor(colorDialog1.Color);

}

}

private void label2\_Click(object sender, EventArgs e)

{

}

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

{

num++;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

list[num].name = "Поток " + (list.Count-1);

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

pictureBox1.BackColor = DefaultColor;

comboBox3.SelectedIndex = 2;

}

private void button2\_Click(object sender, EventArgs e)

{

list[num].stop();

if (list.Count>1)

{

comboBox2.Items.Remove(list[num].name);

list.Remove(list[num]);

if (num > 0)num--;

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void numericUpDown1\_ValueChanged(object sender, EventArgs e)

{

list[num].setSize((int)numericUpDown1.Value);

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

num = comboBox2.SelectedIndex;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

Thread.CurrentThread.Priority = tp[comboBox1.SelectedIndex];

}

private void comboBox3\_SelectedIndexChanged(object sender, EventArgs e)

{

list[num].setPriority(tp[comboBox3.SelectedIndex]);

}

}

public class myThread //класс потока

{

public int size;

public Color color;

private Thread thread;

public ThreadPriority priority;

private Graphics g;

private void loop() {

Random r=new Random((int)(new DateTime()).Ticks);

while (true) {

lock (g)

{

g.FillEllipse((Brush)new SolidBrush(color), r.Next(0, 430 - size), r.Next(0, 415 - size), size, size);

//синхронизация потоков

//Thread.Sleep(1); //чтобы дать форме с низким приоритетом хоть какой то шанс среагировать !!!!!!

}

}

} //главный цикл потока

public myThread(string name,Graphics g,int size,Color color,ThreadPriority initial) {

this.name = name;

this.size = size;

this.color = color;

this.priority = initial;

this.g = g;

thread = new Thread(new ThreadStart(loop));

thread.Start();

} //создание потока

public string name;

public void setColor(Color a) { color = a; }

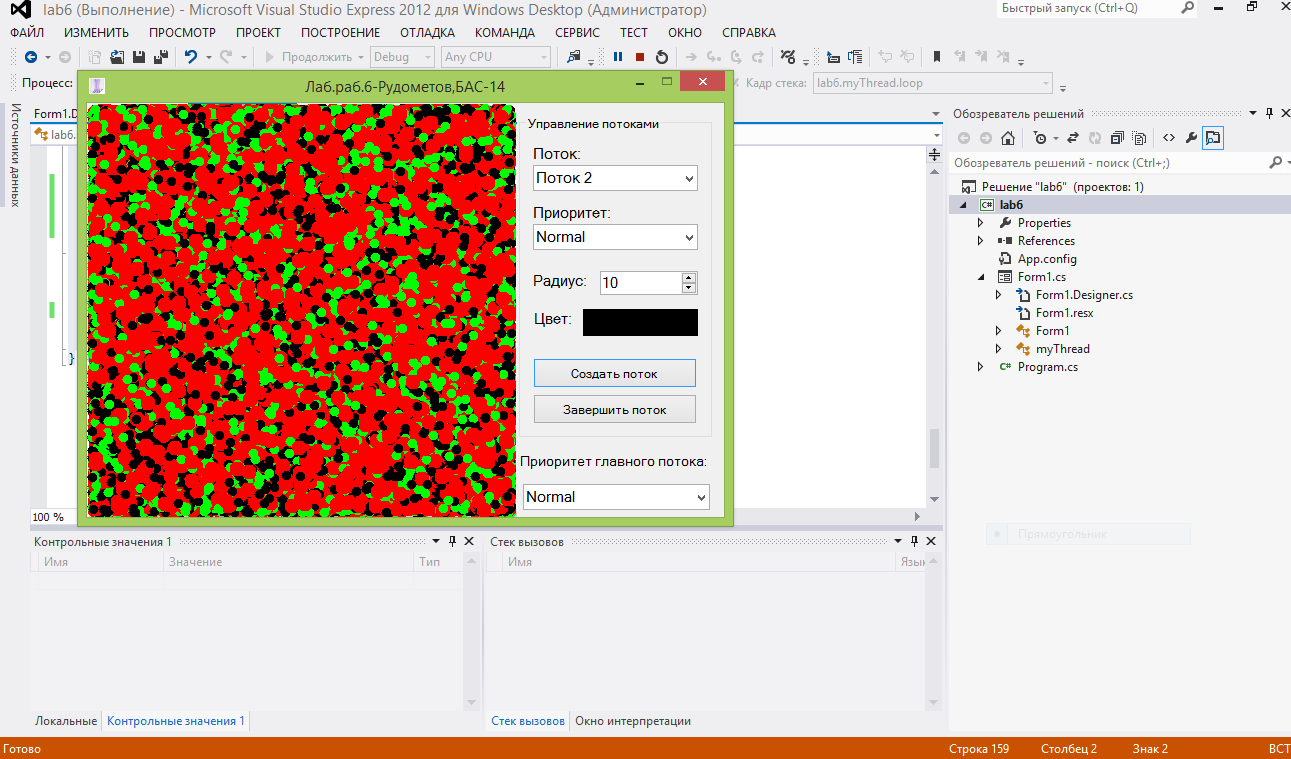
public void setSize(int s) { size = s; }

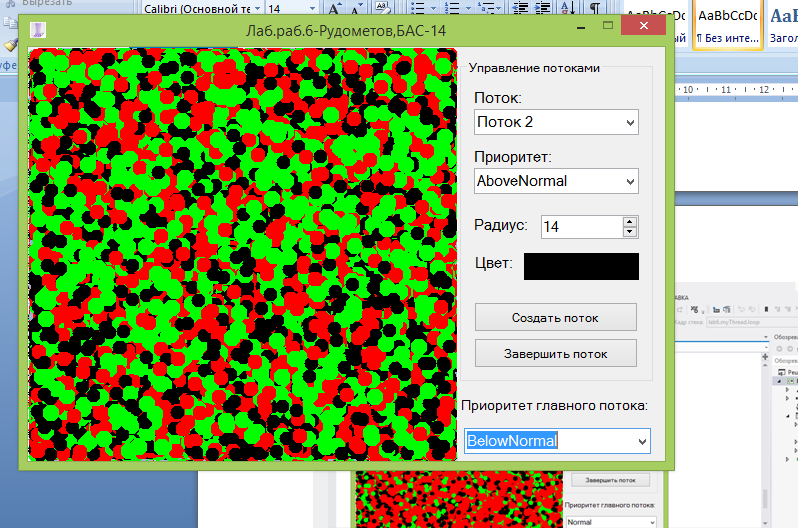
public void setPriority(ThreadPriority t) { priority = t; }

public void stop() { thread.Abort(); }

}

}





Дополнительная информация:

Отчёт

Лабораторная работа 6

Рудометов Глеб БАС-14

**Задание**: Разработать многопоточное приложение, позволяющие динамически управлять потоками, выводящими на экран круги произвольного радиуса и цвета.

Список «Поток:» содержит созданные в программе потоки. Изменение параметров «Приоритет:», «Радиус:» и «Цвет:» применяются к выбранному потоку. Список «Приоритет» позволяет указать приоритет потока

В поле «Радиус:» вводится радиус кругов, рисуемых текущим потоком. При щелчке на поле «Цвет:» выводится стандартное диалоговое окно для выбора цвета потока

Дополнительно на форме должны быть две командные кнопки (рис. 3): 1) при нажатии на кнопку «Создать поток» создается новый поток с параметрами по умолчанию, имя потока добавляется в список потоков; 2) нажатие кнопки «Завершить поток» уничтожает выбранный в списке поток и удаляет его из этого списка. Список «Приоритет главного потока:» позволяет задать приоритет потока формы.

**Текст программы:**

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Threading;

namespace lab6

{

public partial class Form1 : Form

{

ThreadPriority[] tp = {ThreadPriority.Lowest,ThreadPriority.BelowNormal,ThreadPriority.Normal,ThreadPriority.AboveNormal,ThreadPriority.Highest};

List<myThread> list = new List<myThread>();

int num = 0; //номер текущего потока и макс.кол-во

public Graphics g;

Color DefaultColor;

public Form1()

{

InitializeComponent();

g=panel1.CreateGraphics();

DefaultColor = pictureBox1.BackColor;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = 2; //приоритет текущего потока

comboBox1.SelectedIndex = 2; //приоритет главного потока

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

if (colorDialog1.ShowDialog() == DialogResult.OK)

{

pictureBox1.BackColor=colorDialog1.Color;

list[num].setColor(colorDialog1.Color);

}

}

private void label2\_Click(object sender, EventArgs e)

{

}

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

{

num++;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

list[num].name = "Поток " + (list.Count-1);

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

pictureBox1.BackColor = DefaultColor;

comboBox3.SelectedIndex = 2;

}

private void button2\_Click(object sender, EventArgs e)

{

list[num].stop();

if (list.Count>1)

{

comboBox2.Items.Remove(list[num].name);

list.Remove(list[num]);

if (num > 0)num--;

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void numericUpDown1\_ValueChanged(object sender, EventArgs e)

{

list[num].setSize((int)numericUpDown1.Value);

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

num = comboBox2.SelectedIndex;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

Thread.CurrentThread.Priority = tp[comboBox1.SelectedIndex];

}

private void comboBox3\_SelectedIndexChanged(object sender, EventArgs e)

{

list[num].setPriority(tp[comboBox3.SelectedIndex]);

}

}

public class myThread //класс потока

{

public int size;

public Color color;

private Thread thread;

public ThreadPriority priority;

private Graphics g;

private void loop() {

Random r=new Random((int)(new DateTime()).Ticks);

while (true) {

lock (g)

{

g.FillEllipse((Brush)new SolidBrush(color), r.Next(0, 430 - size), r.Next(0, 415 - size), size, size);

//синхронизация потоков

//Thread.Sleep(1); //чтобы дать форме с низким приоритетом хоть какой то шанс среагировать !!!!!!

}

}

} //главный цикл потока

public myThread(string name,Graphics g,int size,Color color,ThreadPriority initial) {

this.name = name;

this.size = size;

this.color = color;

this.priority = initial;

this.g = g;

thread = new Thread(new ThreadStart(loop));

thread.Start();

} //создание потока

public string name;

public void setColor(Color a) { color = a; }

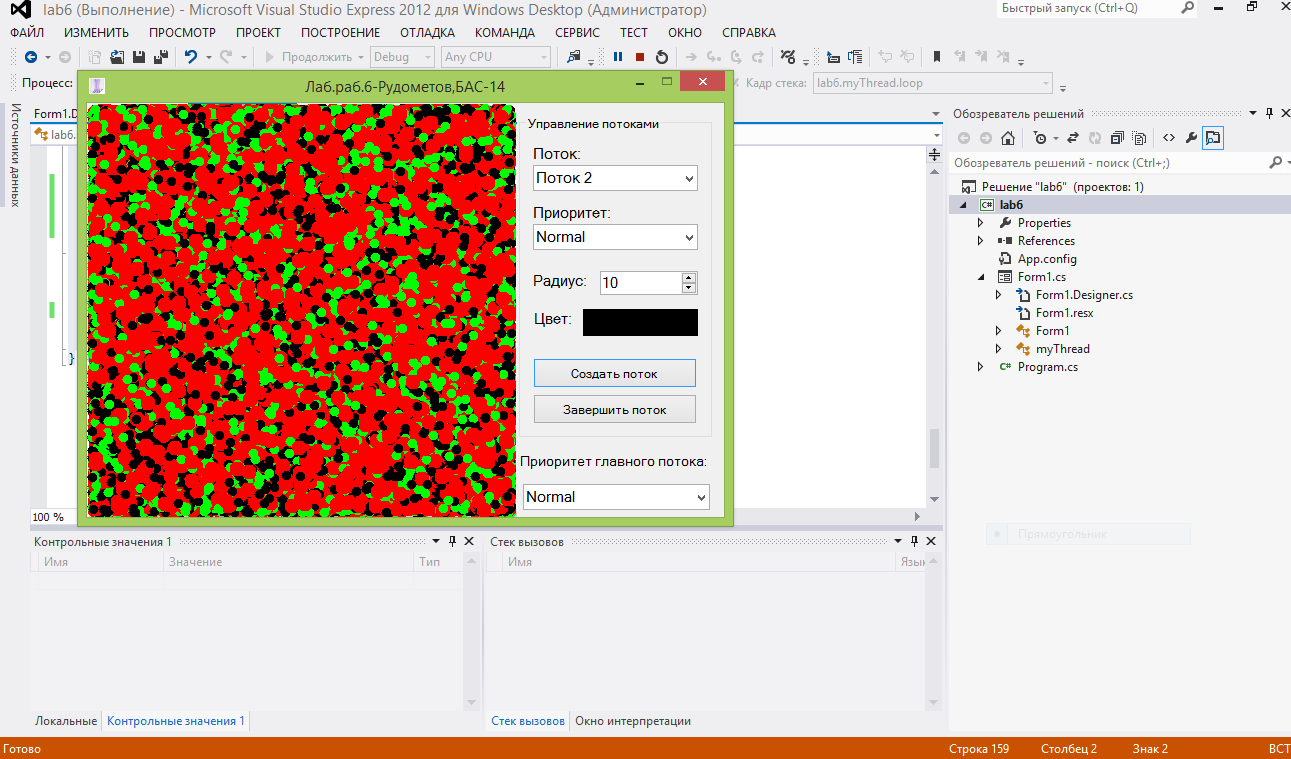
public void setSize(int s) { size = s; }

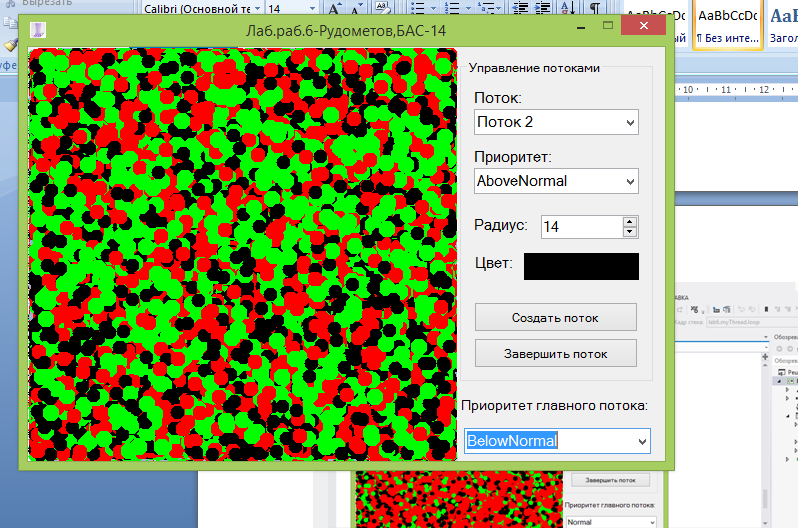
public void setPriority(ThreadPriority t) { priority = t; }

public void stop() { thread.Abort(); }

}

}





Отчёт

Лабораторная работа 6

Рудометов Глеб БАС-14

**Задание**: Разработать многопоточное приложение, позволяющие динамически управлять потоками, выводящими на экран круги произвольного радиуса и цвета.

Список «Поток:» содержит созданные в программе потоки. Изменение параметров «Приоритет:», «Радиус:» и «Цвет:» применяются к выбранному потоку. Список «Приоритет» позволяет указать приоритет потока

В поле «Радиус:» вводится радиус кругов, рисуемых текущим потоком. При щелчке на поле «Цвет:» выводится стандартное диалоговое окно для выбора цвета потока

Дополнительно на форме должны быть две командные кнопки (рис. 3): 1) при нажатии на кнопку «Создать поток» создается новый поток с параметрами по умолчанию, имя потока добавляется в список потоков; 2) нажатие кнопки «Завершить поток» уничтожает выбранный в списке поток и удаляет его из этого списка. Список «Приоритет главного потока:» позволяет задать приоритет потока формы.

**Текст программы:**

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Threading;

namespace lab6

{

public partial class Form1 : Form

{

ThreadPriority[] tp = {ThreadPriority.Lowest,ThreadPriority.BelowNormal,ThreadPriority.Normal,ThreadPriority.AboveNormal,ThreadPriority.Highest};

List<myThread> list = new List<myThread>();

int num = 0; //номер текущего потока и макс.кол-во

public Graphics g;

Color DefaultColor;

public Form1()

{

InitializeComponent();

g=panel1.CreateGraphics();

DefaultColor = pictureBox1.BackColor;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = 2; //приоритет текущего потока

comboBox1.SelectedIndex = 2; //приоритет главного потока

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

if (colorDialog1.ShowDialog() == DialogResult.OK)

{

pictureBox1.BackColor=colorDialog1.Color;

list[num].setColor(colorDialog1.Color);

}

}

private void label2\_Click(object sender, EventArgs e)

{

}

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

{

num++;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

list[num].name = "Поток " + (list.Count-1);

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

pictureBox1.BackColor = DefaultColor;

comboBox3.SelectedIndex = 2;

}

private void button2\_Click(object sender, EventArgs e)

{

list[num].stop();

if (list.Count>1)

{

comboBox2.Items.Remove(list[num].name);

list.Remove(list[num]);

if (num > 0)num--;

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void numericUpDown1\_ValueChanged(object sender, EventArgs e)

{

list[num].setSize((int)numericUpDown1.Value);

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

num = comboBox2.SelectedIndex;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

Thread.CurrentThread.Priority = tp[comboBox1.SelectedIndex];

}

private void comboBox3\_SelectedIndexChanged(object sender, EventArgs e)

{

list[num].setPriority(tp[comboBox3.SelectedIndex]);

}

}

public class myThread //класс потока

{

public int size;

public Color color;

private Thread thread;

public ThreadPriority priority;

private Graphics g;

private void loop() {

Random r=new Random((int)(new DateTime()).Ticks);

while (true) {

lock (g)

{

g.FillEllipse((Brush)new SolidBrush(color), r.Next(0, 430 - size), r.Next(0, 415 - size), size, size);

//синхронизация потоков

//Thread.Sleep(1); //чтобы дать форме с низким приоритетом хоть какой то шанс среагировать !!!!!!

}

}

} //главный цикл потока

public myThread(string name,Graphics g,int size,Color color,ThreadPriority initial) {

this.name = name;

this.size = size;

this.color = color;

this.priority = initial;

this.g = g;

thread = new Thread(new ThreadStart(loop));

thread.Start();

} //создание потока

public string name;

public void setColor(Color a) { color = a; }

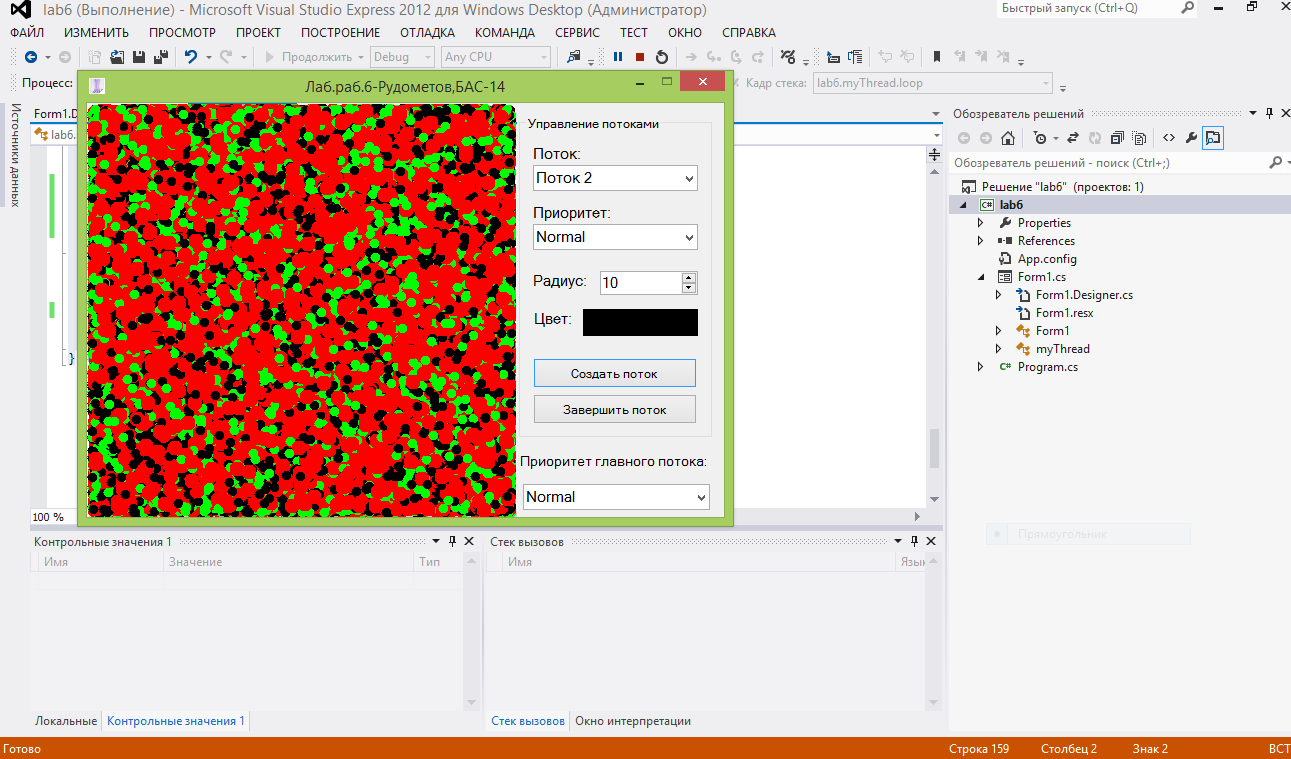
public void setSize(int s) { size = s; }

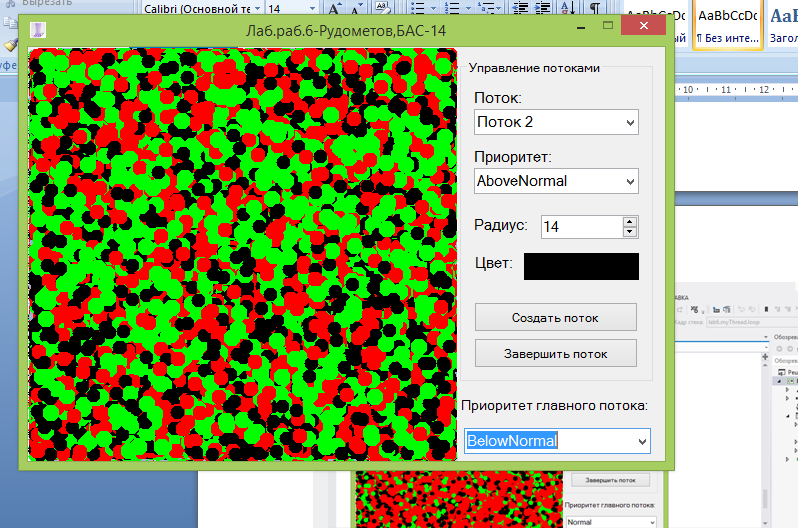
public void setPriority(ThreadPriority t) { priority = t; }

public void stop() { thread.Abort(); }

}

}





Дополнительная информация:

Отчёт

Лабораторная работа 6

Рудометов Глеб БАС-14

**Задание**: Разработать многопоточное приложение, позволяющие динамически управлять потоками, выводящими на экран круги произвольного радиуса и цвета.

Список «Поток:» содержит созданные в программе потоки. Изменение параметров «Приоритет:», «Радиус:» и «Цвет:» применяются к выбранному потоку. Список «Приоритет» позволяет указать приоритет потока

В поле «Радиус:» вводится радиус кругов, рисуемых текущим потоком. При щелчке на поле «Цвет:» выводится стандартное диалоговое окно для выбора цвета потока

Дополнительно на форме должны быть две командные кнопки (рис. 3): 1) при нажатии на кнопку «Создать поток» создается новый поток с параметрами по умолчанию, имя потока добавляется в список потоков; 2) нажатие кнопки «Завершить поток» уничтожает выбранный в списке поток и удаляет его из этого списка. Список «Приоритет главного потока:» позволяет задать приоритет потока формы.

**Текст программы:**

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Threading;

namespace lab6

{

public partial class Form1 : Form

{

ThreadPriority[] tp = {ThreadPriority.Lowest,ThreadPriority.BelowNormal,ThreadPriority.Normal,ThreadPriority.AboveNormal,ThreadPriority.Highest};

List<myThread> list = new List<myThread>();

int num = 0; //номер текущего потока и макс.кол-во

public Graphics g;

Color DefaultColor;

public Form1()

{

InitializeComponent();

g=panel1.CreateGraphics();

DefaultColor = pictureBox1.BackColor;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = 2; //приоритет текущего потока

comboBox1.SelectedIndex = 2; //приоритет главного потока

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

if (colorDialog1.ShowDialog() == DialogResult.OK)

{

pictureBox1.BackColor=colorDialog1.Color;

list[num].setColor(colorDialog1.Color);

}

}

private void label2\_Click(object sender, EventArgs e)

{

}

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

{

num++;

list.Add(new myThread("Поток " + num, g, 10, DefaultColor, tp[2]));

list[num].name = "Поток " + (list.Count-1);

comboBox2.Items.Add(list[num].name);

comboBox2.SelectedIndex = num;

pictureBox1.BackColor = DefaultColor;

comboBox3.SelectedIndex = 2;

}

private void button2\_Click(object sender, EventArgs e)

{

list[num].stop();

if (list.Count>1)

{

comboBox2.Items.Remove(list[num].name);

list.Remove(list[num]);

if (num > 0)num--;

comboBox2.SelectedIndex = num;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void numericUpDown1\_ValueChanged(object sender, EventArgs e)

{

list[num].setSize((int)numericUpDown1.Value);

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

num = comboBox2.SelectedIndex;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

Thread.CurrentThread.Priority = tp[comboBox1.SelectedIndex];

}

private void comboBox3\_SelectedIndexChanged(object sender, EventArgs e)

{

list[num].setPriority(tp[comboBox3.SelectedIndex]);

}

}

public class myThread //класс потока

{

public int size;

public Color color;

private Thread thread;

public ThreadPriority priority;

private Graphics g;

private void loop() {

Random r=new Random((int)(new DateTime()).Ticks);

while (true) {

lock (g)

{

g.FillEllipse((Brush)new SolidBrush(color), r.Next(0, 430 - size), r.Next(0, 415 - size), size, size);

//синхронизация потоков

//Thread.Sleep(1); //чтобы дать форме с низким приоритетом хоть какой то шанс среагировать !!!!!!

}

}

} //главный цикл потока

public myThread(string name,Graphics g,int size,Color color,ThreadPriority initial) {

this.name = name;

this.size = size;

this.color = color;

this.priority = initial;

this.g = g;

thread = new Thread(new ThreadStart(loop));

thread.Start();

} //создание потока

public string name;

public void setColor(Color a) { color = a; }

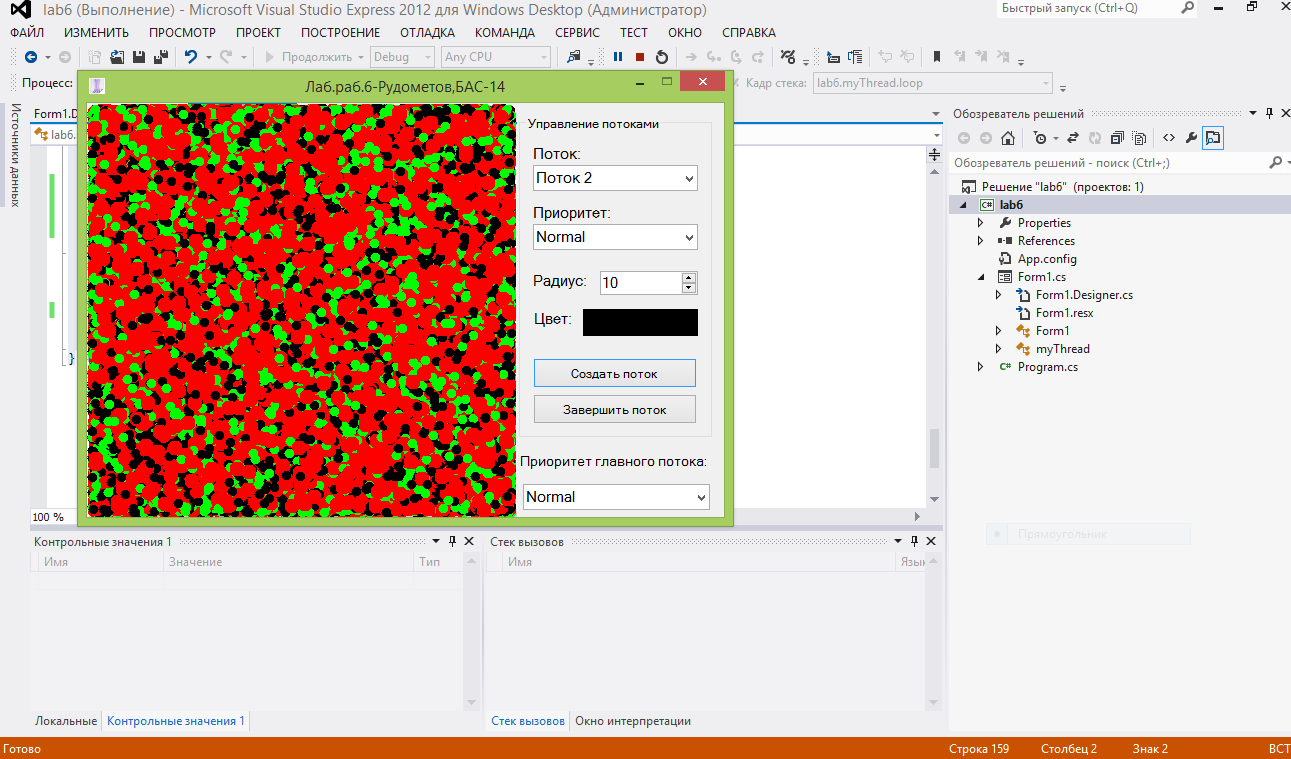
public void setSize(int s) { size = s; }

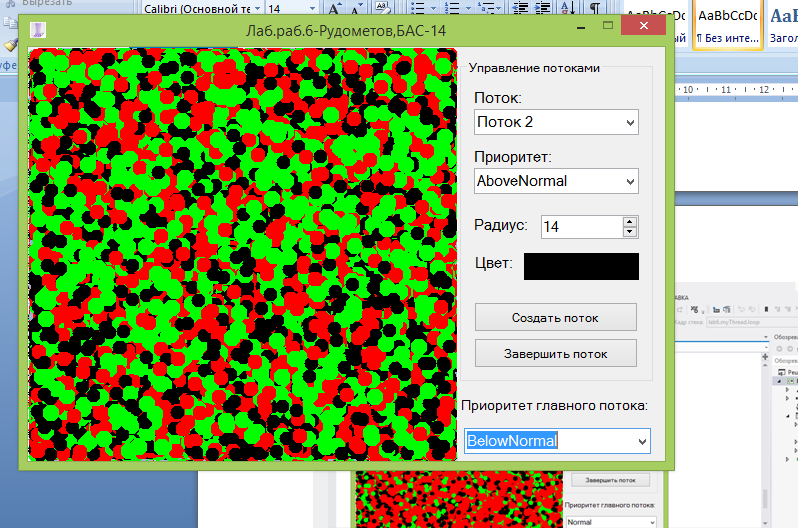
public void setPriority(ThreadPriority t) { priority = t; }

public void stop() { thread.Abort(); }

}

}





private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void numericUpDown1\_ValueChanged(object sender, EventArgs e)

{

list[num].setSize((int)numericUpDown1.Value);

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

num = comboBox2.SelectedIndex;

comboBox3.SelectedIndex = Array.IndexOf(tp, list[num].priority);

pictureBox1.BackColor = list[num].color;

numericUpDown1.Value = list[num].size;

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

Thread.CurrentThread.Priority = tp[comboBox1.SelectedIndex];

}

private void comboBox3\_SelectedIndexChanged(object sender, EventArgs e)

{

list[num].setPriority(tp[comboBox3.SelectedIndex]);

}

}

public class myThread //класс потока

{

public int size;

public Color color;

private Thread thread;

public ThreadPriority priority;

private Graphics g;

private void loop() {

Random r=new Random((int)(new DateTime()).Ticks);

while (true) {

lock (g)

{

g.FillEllipse((Brush)new SolidBrush(color), r.Next(0, 430 - size), r.Next(0, 415 - size), size, size);

//синхронизация потоков

//Thread.Sleep(1); //чтобы дать форме с низким приоритетом хоть какой то шанс среагировать !!!!!!

}

}

} //главный цикл потока

public myThread(string name,Graphics g,int size,Color color,ThreadPriority initial) {

this.name = name;

this.size = size;

this.color = color;

this.priority = initial;

this.g = g;

thread = new Thread(new ThreadStart(loop));

thread.Start();

} //создание потока

public string name;

public void setColor(Color a) { color = a; }

public void setSize(int s) { size = s; }

public void setPriority(ThreadPriority t) { priority = t; }

public void stop() { thread.Abort(); }

}

public myThread(string name,Graphics g,int size,Color color,ThreadPriority initial) {

this.name = name;

this.size = size;

this.color = color;

this.priority = initial;

this.g = g;

thread = new Thread(new ThreadStart(loop));

thread.Start();

} //создание потока

public string name;

public void setColor(Color a) { color = a; }

public void setSize(int s) { size = s; }

public void setPriority(ThreadPriority t) { priority = t; }

public void stop() { thread.Abort(); }

}

public myThread(string name,Graphics g,int size,Color color,ThreadPriority initial) {

this.name = name;

this.size = size;

this.color = color;

this.priority = initial;

this.g = g;

thread = new Thread(new ThreadStart(loop));

thread.Start();

} //создание потока

public string name;

public void setColor(Color a) { color = a; }

public void setSize(int s) { size = s; }

public void setPriority(ThreadPriority t) { priority = t; }

public void stop() { thread.Abort(); }

}

public myThread(string name,Graphics g,int size,Color color,ThreadPriority initial) {

this.name = name;

this.size = size;

this.color = color;

this.priority = initial;

this.g = g;

thread = new Thread(new ThreadStart(loop));

thread.Start();

} //создание потока

public string name;

public void setColor(Color a) { color = a; }

public void setSize(int s) { size = s; }

public void setPriority(ThreadPriority t) { priority = t; }

public void stop() { thread.Abort(); }

}

public myThread(string name,Graphics g,int size,Color color,ThreadPriority initial) {

this.name = name;

this.size = size;

this.color = color;

this.priority = initial;

this.g = g;

thread = new Thread(new ThreadStart(loop));

thread.Start();

} //создание потока

public string name;

public void setColor(Color a) { color = a; }

public void setSize(int s) { size = s; }

public void setPriority(ThreadPriority t) { priority = t; }

public void stop() { thread.Abort(); }

}

public myThread(string name,Graphics g,int size,Color color,ThreadPriority initial) {

this.name = name;

this.size = size;

this.color = color;

this.priority = initial;

this.g = g;

thread = new Thread(new ThreadStart(loop));

thread.Start();

} //создание потока

public string name;

public void setColor(Color a) { color = a; }

public void setSize(int s) { size = s; }

public void setPriority(ThreadPriority t) { priority = t; }

public void stop() { thread.Abort(); }

}

public myThread(string name,Graphics g,int size,Color color,ThreadPriority initial) {

this.name = name;

this.size = size;

this.color = color;

this.priority = initial;

this.g = g;

thread = new Thread(new ThreadStart(loop));

thread.Start();

} //создание потока

public string name;

public void setColor(Color a) { color = a; }

public void setSize(int s) { size = s; }

public void setPriority(ThreadPriority t) { priority = t; }

public void stop() { thread.Abort(); }

}

public myThread(string name,Graphics g,int size,Color color,ThreadPriority initial) {

this.name = name;

this.size = size;

this.color = color;

this.priority = initial;

this.g = g;

thread = new Thread(new ThreadStart(loop));

thread.Start();

}

public string name;

public void setColor(Color a) { color = a; }

public void setSize(int s) { size = s; }

public void setPrioR