Министерство образования и науки Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение науки и высшего образования**

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем и программной инженерии

**Лабораторная работа № 7**

**по дисциплине**

**«Программирование компьютерной графики»**

**Тема: «Алгоритмы обработки растровых изображений»**

Выполнил:

ст. гр. ПРИ-117

Хлызова В.Г.

Принял:

Жигалов И.Е.

Владимир, 2020

ЦЕЛЬ РАБОТЫ

Изучение методов преобразования растровых изображений, приводящих к их визуальному изменению с целью улучшения и получения различных специальных эффектов.

ЗАДАНИЕ

1. Ознакомиться по методическим указаниям и литературе с теоретическим материалом.

2. Выполнить действия, приведенные в разделе 7.2, доработав программу из раздела 6.5. При разработке программы имя проекта, создаваемого в MS Visual Studio, должно содержать фамилию студента и группу (например, Ivanov\_Ivan\_ISG\_105\_lab\_1).

ХОД РАБОТЫ

Добавляем на форму новый раздел меню «Применить фильтр»

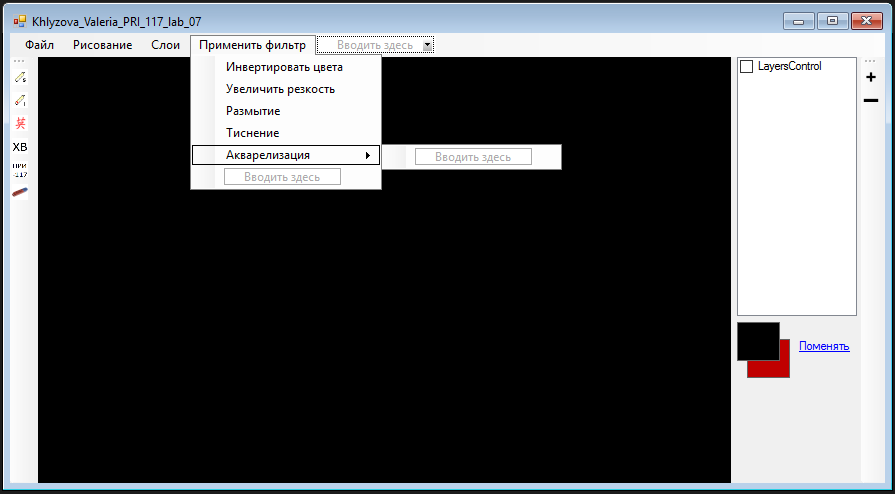


Рисунок 1. Конструктор формы.

**К каждому новому разделу меню добавляем обработчик:**

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

{

ProgrammDrawingEngine.Filter\_0();

}

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

{

ProgrammDrawingEngine.Filter\_1();

}

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

{

ProgrammDrawingEngine.Filter\_2();

}

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

{

ProgrammDrawingEngine.Filter\_3();

}

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

{

ProgrammDrawingEngine.Filter\_4();

}

**В класс anEngine добавляем реализацию фильтров:**

public void Filter\_0()

{

((anLayer)Layers[ActiveLayerNom]).Invers();

}

public void Filter\_1()

{

float[] mat = new float[9]; mat[0] = -0.1f;

mat[1] = -0.1f;

mat[2] = -0.1f;

mat[3] = -0.1f;

mat[4] = 1.8f;

mat[5] = -0.1f;

mat[6] = -0.1f;

mat[7] = -0.1f;

mat[8] = -0.1f;

((anLayer)Layers[ActiveLayerNom]).PixelTransformation(mat, 0, 1, false);

}

public void Filter\_2()

{

float[] mat = new float[9];

mat[0] = 0.05f;

mat[1] = 0.05f;

mat[2] = 0.05f;

mat[3] = 0.05f;

mat[4] = 0.6f;

mat[5] = 0.05f;

mat[6] = 0.05f;

mat[7] = 0.05f;

mat[8] = 0.05f;

((anLayer)Layers[ActiveLayerNom]).PixelTransformation(mat, 0, 1, false);

}

public void Filter\_3()

{

float[] mat = new float[9];

mat[0] = -1.0f;

mat[1] = -1.0f;

mat[2] = -1.0f;

mat[3] = -1.0f;

mat[4] = 8.0f;

mat[5] = -1.0f;

mat[6] = -1.0f;

mat[7] = -1.0f;

mat[8] = -1.0f;

((anLayer)Layers[ActiveLayerNom]).PixelTransformation(mat, 0, 2, true);

}

public void Filter\_4()

{

float[] mat = new float[9];

mat[0] = 0.50f;

mat[1] = 1.0f;

mat[2] = 0.50f;

mat[3] = 1.0f;

mat[4] = 2.0f;

mat[5] = 1.0f;

mat[6] = 0.50f;

mat[7] = 1.0f;

mat[8] = 0.50f;

((anLayer)Layers[ActiveLayerNom]).PixelTransformation(mat, 0, 2, true);

mat[0] = -0.5f;

mat[1] = -0.5f;

mat[2] = -0.5f;

mat[3] = -0.5f;

mat[4] = 6.0f;

mat[5] = -0.5f;

mat[6] = -0.5f;

mat[7] = -0.5f;

mat[8] = -0.5f;

((anLayer)Layers[ActiveLayerNom]).PixelTransformation(mat, 0, 1, false);

}

**В класс anLayer добавляем метод для преобразования PixelTransformation:**

public void PixelTransformation(float[] mat, int corr, float COEFF, bool need\_count\_correction)

{

float[] resault\_RGB = new float[3];

int count = 0;

for (int Y = 0; Y < Heigth; Y++)

{

for (int X = 0; X < Width; X++)

{

for (int c = 0, ax = 0, bx = 0; c < 3; c++)

{

resault\_RGB[c] = 0;

count = 0;

for (bx = -1; bx < 2; bx++)

{

for (ax = -1; ax < 2; ax++)

{

if (X + ax < 0 || X + ax > Width - 1 || Y + bx < 0 || Y + bx > Heigth - 1)

{

resault\_RGB[c] += (float)(DrawPlace[X, Y, c]) \* mat[count] \* COEFF + corr;

count++;

continue;

}

resault\_RGB[c] += (float)(DrawPlace[X + ax, Y + bx, c]) \* mat[count] \* COEFF + corr;

count++;

}

}

}

for (int c = 0; c < 3; c++)

{

if (count != 0 && need\_count\_correction)

{

resault\_RGB[c] /= count;

}

if (resault\_RGB[c] < 0)

{

resault\_RGB[c] = 0;

}

if (resault\_RGB[c] > 255)

{

resault\_RGB[c] = 255;

}

DrawPlace[X, Y, c] = (int)resault\_RGB[c];

}

}

}

}

Пример работы программы

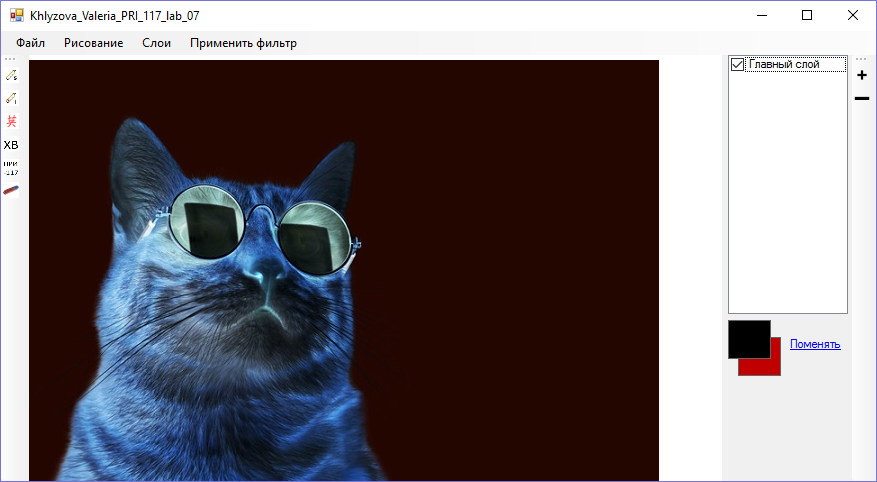


Рисунок 2. Инвертирование цветов.

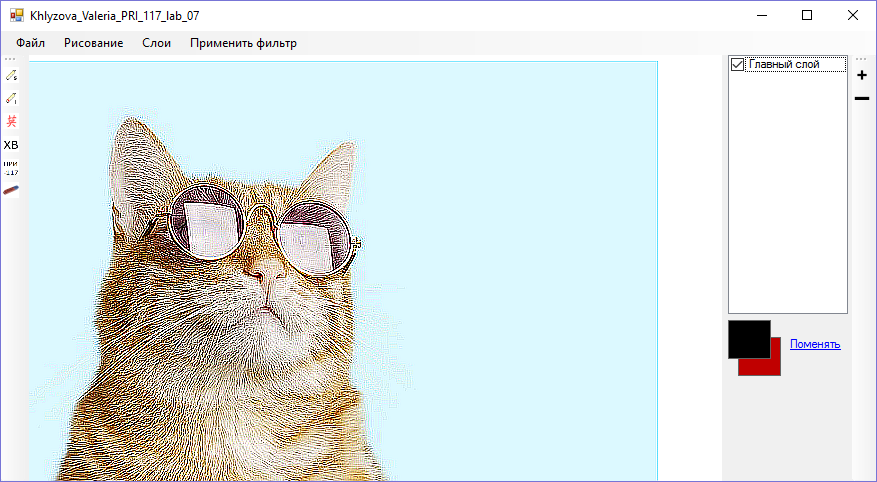


Рисунок 3. Увеличение резкости.

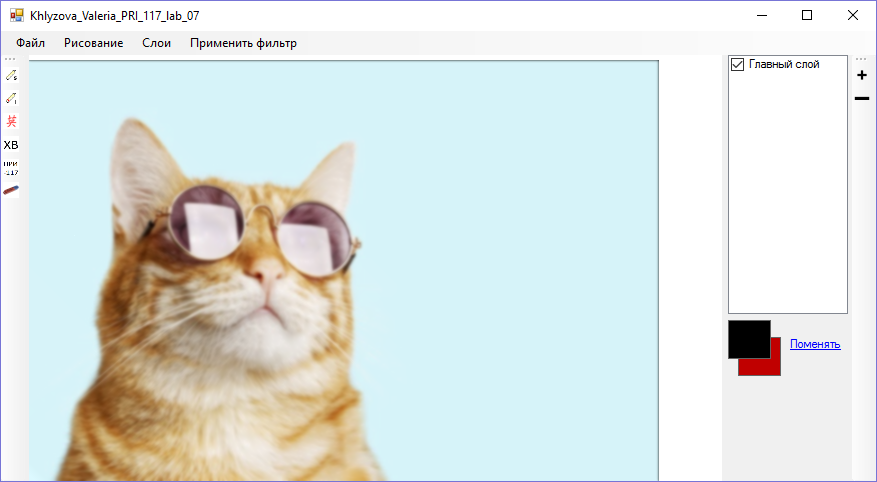


Рисунок 4. Размытие.

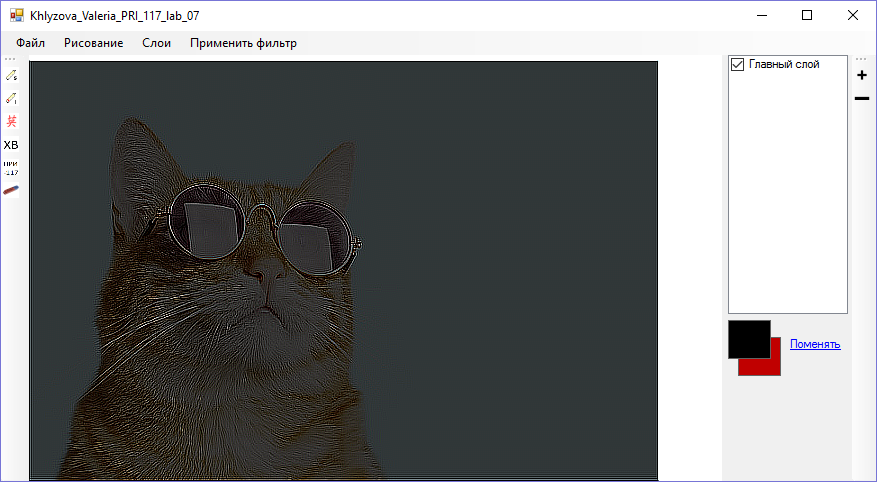


Рисунок 5. Тиснение.



Рисунок 6. Акварелизация.

ВЫВОД

В ходе выполнения лабораторной работы были изучены методы преобразования растровых изображений, приводящих к их визуальному изменению с целью улучшения и получения различных специальных эффектов.