

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

**Курсовая работа по дисциплине
«Программирование» Тема: работа с .png
файлами**

Студент гр.8304

Холковский К. В.

Преподаватель

Чайка К. В.

**Санкт-Петербург
2019**

Задание на курсовую работу

Студент Холковский К. В.

Группа 8304

Тема работы: работа с .png файлами

Исходные данные:

Написать программу для обработки .png файлов с GUI, создать собственный класс для хранения изображения и реализовать методы: Рисование квадрата, рисование окружности, фильтр RGB, разделение изображения на $n*m$ частей.

Содержание пояснительной записки:

- Содержание
- Введение
- MainWindow
- Picture
- Окна для работы с подзадачами
- Image
- Тестирование программы
- Исходный код
- Использованные источники

Дата выдачи задания: 07.03.2019

Дата защиты:

Студент

Преподаватель

Холковский К. В.

Чайка К. В.

АННОТАЦИЯ

В данной работе была создана программа, являющаяся desktop приложением для работы с файлами-изображениями формата png. Был разработан GUI для загрузки/сохранения файлов, рисования круга или квадрата, создания rgb-компонента и разделения изображения. Проведена работа по форматированию кода и предоставлено тестирование программы.

SUMMARY

In this paper, a program was created that is a desktop application for working with image files in png format. A GUI was developed to load / save files, draw a circle or square, create an rgb component and image split. Work was done on formatting the code and provided testing program. Work was done on formatting the code and provided testing program.

Оглавление

Введение	5
MainWindow	6
Picture.....	9
Dialog	11
Image.....	14
Тестирование программы	15
Заключение	18
Список использованных источников	18

Введение.

Приложение написано на языке C++ с использованием фреймворка Qt. Для работы с .png был создан класс Image, алгоритмы для работы с png были написаны без использования средств Qt

Цель и условие работы.

Вариант 17

Общие сведения

- Формат картинки PNG (рекомендуем использовать библиотеку libpng файл всегда соответствует формату PNG)
- обратите внимание на выравнивание; мусорные данные, если их необходимо дописать в файл для выравнивания, должны быть нулями.
- все поля стандартных PNG заголовков в выходном файле должны иметь те же значения что и во входном (разумеется кроме тех, которые должны быть изменены).

Программа должна реализовывать следующий функционал по обработке PNG-файла

Рисование окружности. Окружность определяется:

- **либо** координатами левого верхнего и правого нижнего угла квадрата, в который она вписана, **либо** координатами ее центра и радиусом
- толщиной линии окружности
- цветом линии окружности
- окружность может быть залитой или нет
- цветом которым залита сама окружность, если пользователем выбрана залитая окружность

Фильтр rgb-компонент. Этот инструмент должен позволять для всего изображения либо установить в 0 либо установить в 255 значение заданной компоненты. Функционал определяется

- Какую компоненту требуется изменить
- В какой значение ее требуется изменить

Разделяет изображение на N*M частей. Реализация: либо провести линии заданной толщины, тем самым разделив изображение либо сохранение каждой части в отдельный файл. -- по желанию студента (можно и оба варианта). Функционал определяется:

- Количество частей по “оси” Y
- Количество частей по “оси” X
- Толщина линии
- Цвет линии
- Либо путь куда сохранить кусочки

Рисование квадрата с диагоналями. Квадрат определяется:

- Координатами левого верхнего угла
- Размером стороны
- Толщиной линий
- Цветом линий
- Может быть залит или нет (диагонали располагаются “поверх” заливки)
- Цветом которым он залит, если пользователем выбран залитый

1. MainWindow.

В функции `main()` создается экземпляр класса `MainWindow`. Далее запускается метод `exec()`.

```
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();

    return a.exec();
}
```

Класс `MainWindow` является основообразующим классом в данном проекте. Он предназначен для создания общего интерфейса программы, взаимодействия с классом `Dialog`, который предназначен для создания диалоговых окон и передаче данных из них в класс `Painter`, а также для работы с кнопками, созданными с `.ui` файле.

Конструктор и деструктор класса `MainWindow` :

```
MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    label = new picture;
    setCentralWidget(label);
    label->setAlignment(Qt::AlignLeft|Qt::AlignTop);
}

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

Класс также содержит слоты, реализующие меню (открытие/закрытие, создание и сохранение файла, вывод справочной информации и т.п.). Для открытия файла используется `QFileDialog`, для создания нового файла и сохранения используются методы `Image`.

```
void MainWindow::on_open_triggered() {
    QString filePath = QFileDialog::getOpenFileName(this, "Open
Image", "/home/gagask/CoursWork/", "*.png");
}
```

```

if(!filePath.isEmpty()){
    isopen=true;
    info="Путь к файлу: "+filePath+"\n";
    label->getImage()->read(filePath);
    label->setPixmap(label->getImage()->getPixmap());
    label->resize(label->getImage()->getwidth(),label->getImage()->getheight());
}
}

void MainWindow::on_save_triggered() {
    if(!isopen){
        QMessageBox::critical(nullptr, "Warning!", "Сначала откройте картинку");
        return;    }

    QString save_filename = QFileDialog::getSaveFileName(this, "Choose a path to save
image", "/home/", "*.png");

    if(!save_filename.isEmpty())    label->getImage()->write(save_filename);
}

void MainWindow::on_filter_triggered() {
    if(!isopen){
        QMessageBox::critical(nullptr, "Warning!", "Сначала откройте картинку");
        return;    }

    size_t num;
    QString comp;
    winfilter my;
    my.setModal(true);
    if(!my.exec()) return;
    my.load(&comp, &num);
    size_t a;
    if(comp.toLatin1().constData()[0]=='R') a=0;
    if(comp.toLatin1().constData()[0]=='G') a=1;
    if(comp.toLatin1().constData()[0]=='B') a=2;
    label->getImage()->filterrrgb(a,num);
    label->setPixmap(label->getImage()->getPixmap());
}

void MainWindow::on_circle_triggered() {    if(!isopen){
    QMessageBox::critical(nullptr, "Warning!", "Сначала откройте картинку");
    return;    }

    size_t flag1,fatness,flag2;

```

```

    QColor color2,color1;

    Circle my;
my.setModal(true);

    if(!my.exec()) return;

    my.load(&flag1,&fatness,&color1,&color2,&flag2);

    label->ActivateDoingCircle(flag1,fatness,color1,flag2,color2);
}

void MainWindow::on_squaer_triggered() {
    if(!isopen){
QMessageBox::critical(nullptr, "Warning!", "Сначала откройте картинку");

        return;    }

    size_t l,fatness;

    bool flag;

    QColor color1,color2;

    square my;

    my.setModal(true);

    if(!my.exec()) return;

my.load(&l,&fatness,&color1,&flag,&color2);

    label->ActivateDoingSquare(l,fatness,color1,flag,color2);
}

void MainWindow::on_division_triggered() {
    if(!isopen){
QMessageBox::critical(nullptr, "Warning!", "Сначала откройте картинку");

        return;    }

    int n,m,fat;

    bool flag;

QColor color;

    QString path;

    Division my;

    my.setModal(true);

    if(!my.exec())return;

    my.load(&n,&m,&color,&fat,&flag,&path);

    label->getImage()->division(n,m,color,fat,flag,path);

    label->setPixmap(label->getImage()->getPixmap());
}

void MainWindow::on_action_triggered() {

```



```

if(!isopen){

    QMessageBox::information(this,"Информация","Файл не открыт");

    return;    }

    info+="Ширина: "+QString::number(label->getImage()->getwidth())+"    Высота:
"+QString::number(label->getImage()->getheight())+"\n";

    QMessageBox::information(this,"Информация",info);

}

void MainWindow::on_action_2_triggered() {
QMessageBox::information(this,"Автор","Хол
ковский Константин Владимирович, студент
гр. 8304 СПбГЭТУ (ЛЭТИ) \nВариант курсовой
на 5: 17");
}

```

В данном классе (как и во всех далее) в деструкторе происходит освобождение выделенной памяти.

2. Picture

Это один из основных классов данного проекта в котором хранится используемый объект класса Image. Он отвечает за хранение изображения и рисования на нем. Также при помощи метода setPixmap() было загружено изображение, содержащееся в классе Image. Метод **mousePressEvent(QMouseEvent *event)** отвечает за активацию отлавливания кликов мыши при помощи метода QMouseEvent, а в дальнейшем за требуемое изменение изображения.

```

picture::picture() { }

void picture::mousePressEvent(QMouseEvent* event){

    int l;

    if (event->button()==Qt::LeftButton){

        if(dsquare){

            img.square(event->x(),event-
>y(),squ.l,squ.fatness,squ.color1,squ.flag,squ.color2);

            setPixmap(img.getPixmap());

        }

        if(dcircle){

            if(count==0){

                count++;

                tmp=event->pos();

            }

        }

    }

    else{

```

```

        count=0;

        if (squ.l){

            if (abs(event->x()-tmp.x())>abs(event->y()-tmp.y()))

l=abs(event->y()-tmp.y())/2;

                else l=abs(event->x()-tmp.x())/2;

            if(event->x()>tmp.x())

                if(event->y()>tmp.y())

                    img.circle(tmp.x()+l,tmp.y()+l,l,squ.fatness,squ.color1,squ.flag,squ.color2);

                else

                    img.circle(tmp.x()+l,tmp.y()-l,l,squ.fatness,squ.color1,squ.flag,squ.color2);

                else

                    if(event->y()>tmp.y())

                        img.circle(tmp.x()-l,tmp.y()+l,l,squ.fatness,squ.color1,squ.flag,squ.color2);

                    else

                        img.circle(tmp.x()-l,tmp.y()-

l,l,squ.fatness,squ.color1,squ.flag,squ.color2);

        }

        else{

            l=sqrt((event->x()-tmp.x())*(event->x()-

tmp.x())+(event->y()-tmp.y())*(event->y()-tmp.y()));

            img.circle(tmp.x(),tmp.y(),l,squ.fatness,squ.color1,squ.flag,squ.color2);}

        setPixmap(img.getPixmap());}

        }

        image *picture::getImage() {

            return &img;

        }

        void picture::ActivateDoingSquare(int l,int fat,QColor a,bool f,QColor b) {

            squ.l=l;

            squ.fatness=fat;

            squ.color1=a;

            squ.color2=b;

            squ.flag=f;

            dcircle=false;

            dsquare=true;

        }

        Void picture::ActivateDoingCircle(int flag,int fat, QColor a, bool f, QColor b) {

            squ.l=flag;

            squ.fatness=fat;

            squ.color1=a;

            squ.color2=b;

            squ.flag=f;

```

```
dcircle=true;
dsquare=false; }
```

3. Dialog

Для выбора опций в подзадачах реализованы дополнительные окна в приложении. Для этого были реализованы классы circle, division, winfilter, square наследованные от класса QDialog. Для разных подзадач реализованы собственные диалоговые окна. В общем случае, каждое окно содержит элементы Qt, а также кнопки “Ok” и “Cancel”. А также методы возвращающие значения введенных данных.

Winfilter:

```
winfilter::winfilter(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::winfilter)
{
    ui->setupUi(this);
    ui->comboBox->addItem("R");
    ui->comboBox->addItem("G");
    ui->comboBox->addItem("B");
    ui->radioButton->setChecked(true);
}

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

void winfilter::load(QString* comp, size_t* num)
{
    if (ui->radioButton->isChecked()) *num=255;
    else *num=0;
    *comp=ui->comboBox->currentText();
    return;
}
```

Circle:

```
Circle::Circle(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::Circle)
{
    ui->setupUi(this);
    color_1=Qt::black;
    color_2=Qt::red;
```

```

    ui->spinBox->setMaximum(50);
    ui->spinBox->setMinimum(1);
    ui->radioButton->setChecked(true);
}
Circle::~Circle()
{
    delete ui;
}
void Circle::on_pushButton_clicked()
{
    color_1=QColorDialog::getColor(Qt::black,this,"Chose color lines");
}
void Circle::on_pushButton_2_clicked()
{
    color_2=QColorDialog::getColor(Qt::red,this,"Chose color fill");
}
void Circle::load(size_t* flag,size_t* fatness,QColor* color_line,QColor*
color_inside,size_t* flag_2)
{
    if (ui->radioButton->isChecked()) *flag=0;
    else *flag=1;
    *fatness=ui->spinBox->value();
    *color_line=color_1;
    *color_inside=color_2;
    if (ui->checkBox->isChecked()) *flag_2=1;
    else *flag_2=0;
}

```

Square:

```

square::square(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::square)
{
    ui->setupUi(this);
    ui->spinBox_2->setMaximum(50);
    ui->spinBox_2->setMinimum(1);
    ui->spinBox->setMaximum(300);
    ui->spinBox->setMinimum(1);
}

```

```

    color_1=Qt::black;
    color_2=Qt::red;
}
square::~~square()
{
    delete ui;
}
void square::on_pushButton_clicked()
{
    color_1=QColorDialog::getColor(Qt::black,this,"Chose color lines");
}
void square::on_pushButton_2_clicked()
{
    color_2=QColorDialog::getColor(Qt::black,this,"Chose color fill");
}
void square::load(size_t* l,size_t* fatness,QColor* color1,bool* flag,QColor* color2)
{
    *l=ui->spinBox->value();
    *fatness=ui->spinBox_2->value();
    *color1=color_1;
    if (ui->checkBox->isChecked()) *flag=true;
    else *flag=false;
    *color2=color_2;
}

```

Division:

```

Division::Division(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::Division)
{
    ui->setupUi(this);
    ui->spinBox_3->setMinimum(1);
    ui->spinBox_3->setMaximum(50);
    ui->radioButton->setChecked(true);
    ui->spinBox->setMinimum(1);
    ui->spinBox->setMaximum(50);
    ui->spinBox_2->setMinimum(1);
    ui->spinBox_2->setMaximum(50);
}

```

```

        color=Qt::black;
        file_path = "/home/gagask/CoursWork";
    }
Division::~Division()
{
    delete ui;
}

void Division::load(int *n, int *m, QColor *color,int* fatness, bool *flag,QString*
path)
{
    if(ui->radioButton->isChecked()) *flag = true;
    else *flag = false;
    *n=ui->spinBox_2->value();
    *m=ui->spinBox->value();
    *color=this->color;
    *fatness=ui->spinBox_3->value();
    *path=file_path;
    if(path->isEmpty()) *path="/home/gagask/CoursWork";
}

void Division::on_pushButton_clicked()
{
    color=QColorDialog::getColor(Qt::black,this,"Chose color lines");
}

void Division::on_pushButton_2_clicked()
{
    file_path=QFileDialog::getExistingDirectory(this,"Выберите папку","/home/gagask/
CoursWork");
}

```

4. Image

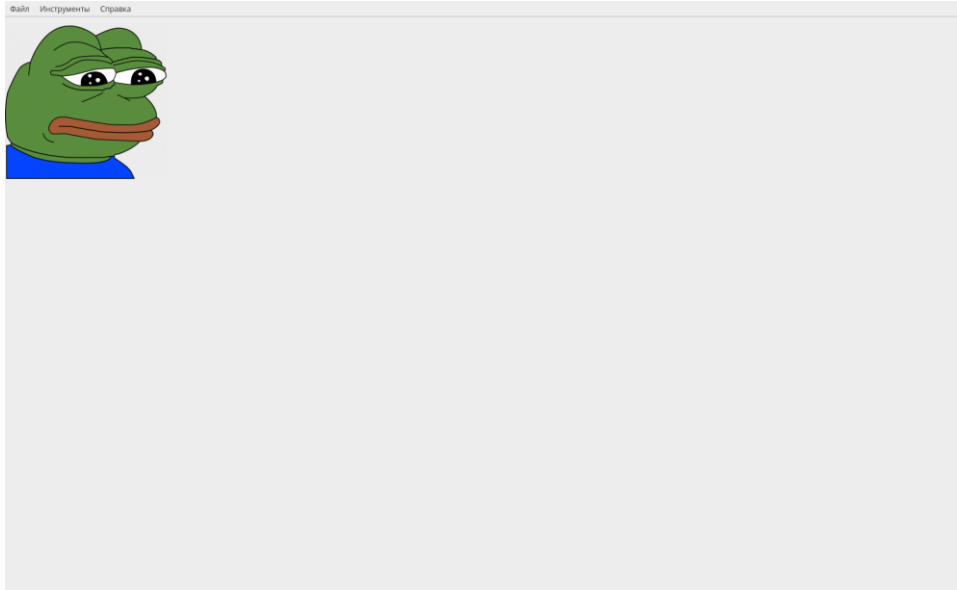
Данный класс предназначен для хранения информации о png файле и решения основных задач: рисования окружности - метод circle, рисования квадрата - square, деления изображения - division, фильтр RGB – filterrgb, считывание изображения - read(используя libpng), сохранение изображения - write(используя libng). Так же методы для отображения изображения - getHight, getWidth,getPixmap.

Тестирование программы

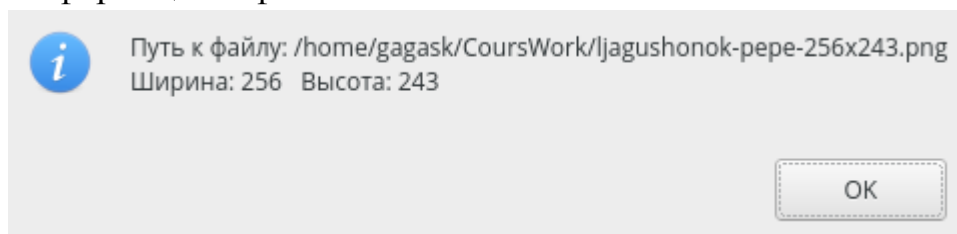
Исходный файл:



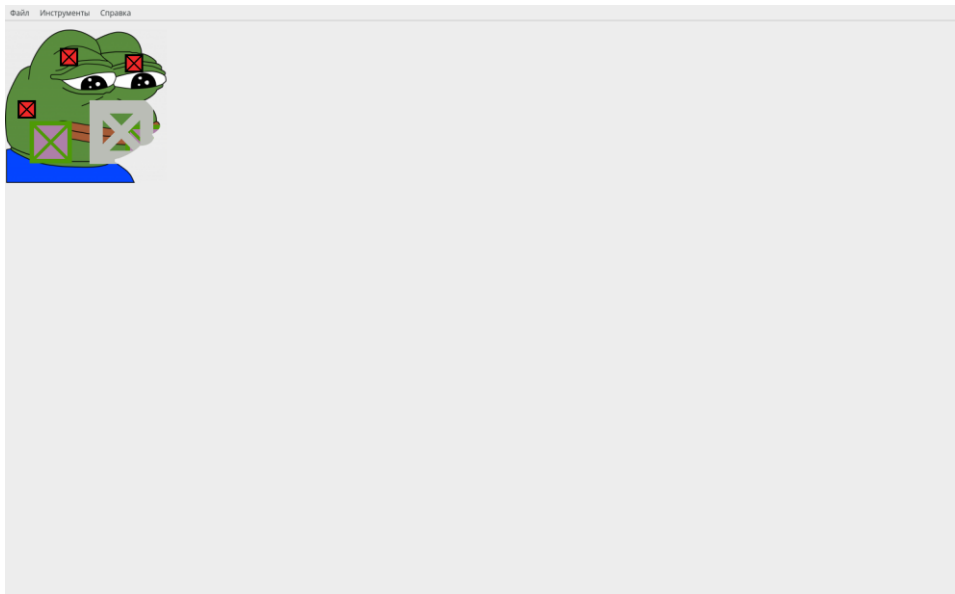
Результат загрузки файла в программу:



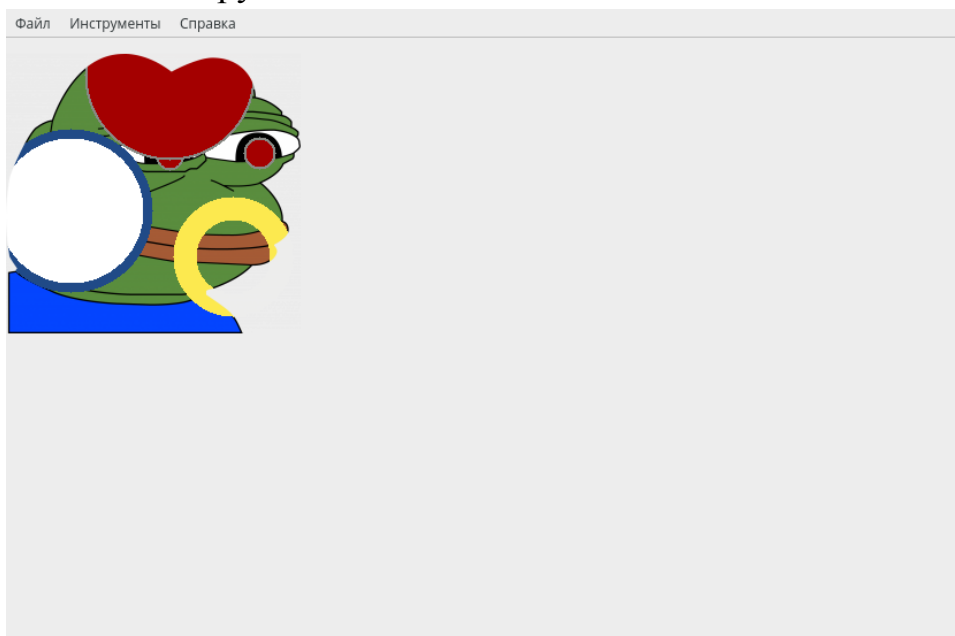
Информация о файле:



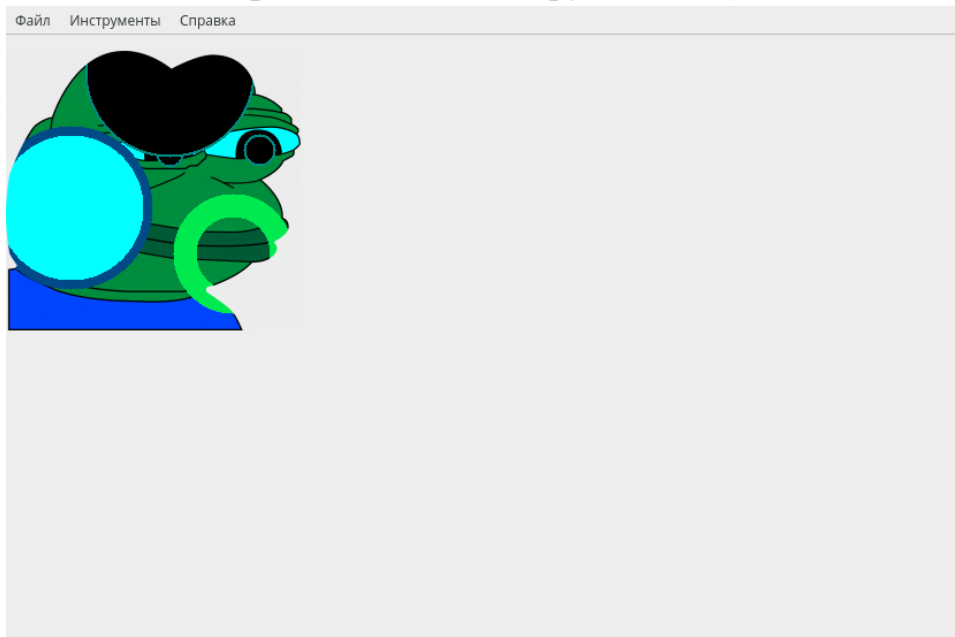
Рисование квадрата:



Рисование окружности:

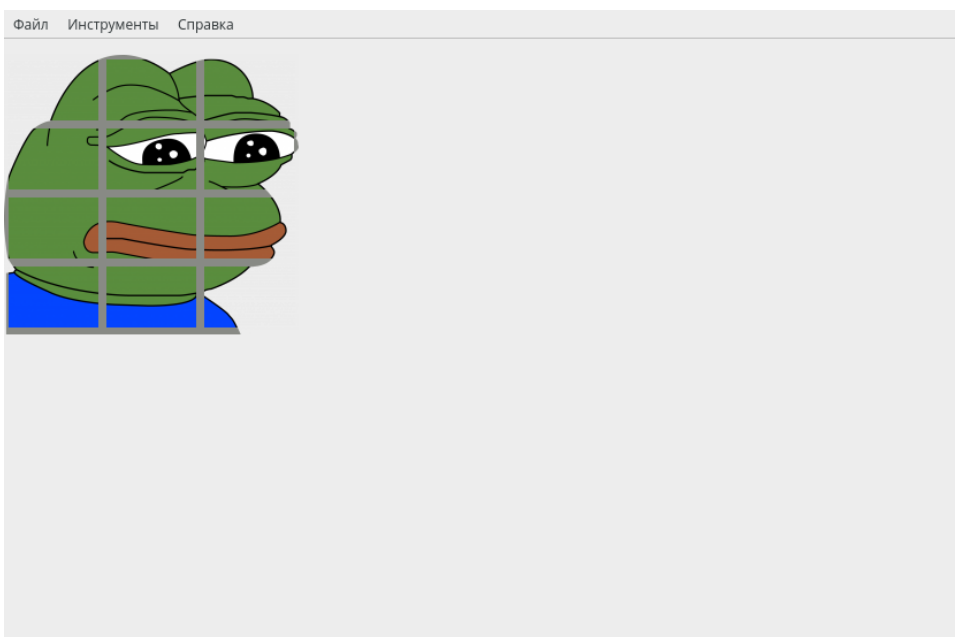


Изменение выбранного цвета на другой(R в 0):

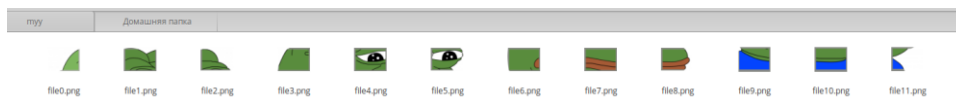


Разделение изображения:

1)Рисуя:



2)Сохраняя по пути:



Заключение. В ходе выполнения работы было создано desktop приложение с GUI для обработки файлов-изображений в формате png. Для работы программы были созданы и описаны все необходимые классы и структуры. Программа была протестирована, результат работы программы соответствует заданным условиям.

Список использованных источников.

1. <https://ru.wikipedia.org/wiki/>
2. Документация фреймворка Qt.
3. <https://www.cyberforum.ru/>
4. <https://prog-cpp.ru/>
5. <https://doc.qt.io/>