

```
//UWAGA: Testowane na Qt Creator 4.2.1 based on Qt 5.8.0

#include <QCoreApplication>
#include <QtCore/QCoreApplication>
#include <QDebug>
#include <QString>
#include <QFile>
#include <QImage>
#include <QColor>
#include "qpainter.h"
#include <QPainter>
#include <QPixmap>

#include <algorithm>

#include <iostream>
#include <sstream>

#include <QApplication>

#include "iostream"

using namespace std;

int mnoz(int & zmienna, int ile)
{
    qDebug()<<"mnoz: zmienna="<<zmienna<<", ile="<<ile;
    zmienna=ile*zmienna;
    qDebug()<<"mnoz: zmienna="<<zmienna;
}

void rys_linie(QImage &img)
{
    int img_width=img.width();
    int img_height=img.height();

    int y=img_height-1;
    for(int x=0; x<img_width; x++)
    {
        if(y>=0) {
            QColor col(255,255,0);
            img.setPixel(x,y,col.rgba()); }
        y--;
    }
}
```

```
int main(int argc, char *argv[])
{
    //QCoreApplication a(argc, argv);
    QApplication a(argc, argv);

    //UWAGA: do zmiany:
    QString path="/sciezka_do_katalogu_z_obrazkami/";

    qDebug()<<" ssss="<<path;
    //QString file_input= path+"TOPRO_cat_N_s1.bmp";
    //    QString file_output=path+"TOPRO_cat_N_s1_output.bmp";

    QString file_input= path+"lena512.bmp";
    QString file_output=path+"lena512_output.bmp";

    qDebug()<<" file_input ="<<file_input ;

    QImage img;
    if(img.load(file_input))
    {
        qDebug()<<"OK. Wczytano obrazek "+ file_input;
    }
    else
    {
        qDebug()<<"BLAD: Nie wczytano obrazka "+ file_input;
        exit(0);
    }

    //odczytujemy szerokosc i wysokosc obrazka
    int img_width=img.width();
    int img_height=img.height();

    qDebug()<<" img_width="<<img_width<<" , img_height="<<img_height;

    //img_output to bedzie wynikowy obrazek
    QImage img_output(img_width,img_height,QImage::Format_RGB32);//
    img_output=img_output(img_width,img_height); //

    //kopiowanie obrazka:
    for(int x=0; x<img_width; x++)
        for(int y=0; y<img_height; y++)
        {
            QColor col(img.pixel(x,y));
            //qDebug()<< col.red();
            img_output.setPixel(x,y,col.rgb());
        }
}
```

```
//rysowanie czerw. kreski
for(int x=0; x<img_width*3/4; x++)
{
    QColor col(255,0,0);
    img_output.setPixel(x,x,col.rgb());
}

//dla x i y
for(int x=0.375*img_width; x<0.625*img_width; x++)
{
    for(int y=0.375*img_height; y<0.625*img_height; y++)
    {
        QColor col(img.pixel(x,y));
        col.setBlue(255);
        img_output.setPixel(x,y,col.rgb());
    }
}

//napis
QPainter p(&img_output);
p.setFont(QFont("Times", 12));
p.setPen(QPen(Qt::green));
QPoint pkt(img_width/20, img_height/4);
p.drawText(pkt, "Testowy napis");

//zapis obrazka
QDebug() << "Zapisujemy " << file_output;
img_output.save(file_output);

// QString file_input2= path+"war-time-1.bmp";
// QString file_output2=path+"war-time-1_output.bmp";

QString file_input2= path+"car.bmp";
QString file_output2=path+"car_output.bmp";

QImage img2;
if(img2.load(file_input2))
{
    qDebug() << "OK. Wczytano obrazek "+ file_input2;
}
else
{
```

```
    qDebug()<<"BLAD: Nie wczytano obrazka "+ file_input2;
    exit(0);
}

//odczytujemy szerokosc i wysokosc obrazka
int img2_width=img2.width();
int img2_height=img2.height();

QImage img2_output(img2_width,img2_height,QImage::Format_RGB32);//
img_width,img_height); //

int val_min=255, val_max=0;

for(int x=0; x<img2_width; x++)
    for(int y=0; y<img2_height; y++)
    {
        QColor col(img2.pixel(x,y));
        //qDebug()<< col.red();
        // img_output.setPixel(x,y,col.rgba());
        if(col.red()<val_min) {val_min=col.red();}
        if(col.red()>val_max) {val_max=col.red();}
    }

qDebug()<<" val_min="<<val_min<<" , val_max="<<val_max;

for(int x=0; x<img2_width; x++)
    for(int y=0; y<img2_height; y++)
    {
        QColor col(img2.pixel(x,y));
        //qDebug()<< col.red();
        int cc=col.red();

        cc=255*(cc-val_min)/(val_max-val_min);

        QColor col2(cc,cc,cc);

        img2_output.setPixel(x,y,col2.rgba());
    }

qDebug()<< "Zapisujemy " << file_output2;
img2_output.save(file_output2);

//*****

QString file_input3= path+"tmp_col.bmp";
QString file_output3=path+"tmp_col_output.bmp";
```

```
QImage img3;
img3.load(file_input3);

int img3_width=img3.width();
int img3_height=img3.height();

QImage img3_output(img3_width,img3_height,QImage::Format_RGB32);//
img_width,img_height); //

int val_r_min=255, val_r_max=0;
int val_g_min=255, val_g_max=0;
int val_b_min=255, val_b_max=0;

for(int x=0; x<img3_width; x++)
    for(int y=0; y<img3_height; y++)
    {
        QColor col(img3.pixel(x,y));
        //qDebug()<< col.red();
        // img_output.setPixel(x,y,col.rgba());
        if(col.red()<val_r_min) {val_r_min=col.red();}
        if(col.red()>val_r_max) {val_r_max=col.red();}

        if(col.green()<val_g_min)
{val_g_min=col.green();}
        if(col.green()>val_g_max)
{val_g_max=col.green();}

        if(col.blue()<val_b_min)
{val_b_min=col.blue();}
        if(col.blue()>val_b_max)
{val_b_max=col.blue();}

    }

qDebug()<<" r_min="<<val_r_min<<" r_max="<<val_r_max;
qDebug()<<" g_min="<<val_g_min<<" g_max="<<val_g_max;
qDebug()<<" b_min="<<val_b_min<<" b_max="<<val_b_max;

for(int x=0; x<img3_width; x++)
    for(int y=0; y<img3_height; y++)
    {
        QColor col(img3.pixel(x,y));
        //qDebug()<< col.red();
        int cr=col.red();
        int cg=col.green();
        int cb=col.blue();
```

```
        cr=255*(cr-val_r_min)/(val_r_max-val_r_min);
        cg=255*(cg-val_g_min)/(val_g_max-val_g_min);
        cb=255*(cb-val_b_min)/(val_b_max-val_b_min);

        QColor col2(cr,cg,cb);

        img3_output.setPixel(x,y,col2.rgb());
    }

    rys_linie(img3_output);

    qDebug()<< "Zapisujemy " << file_output3;
    img3_output.save(file_output3);

    std::pair<int,QString> moja_para;
    moja_para.first=1;
    moja_para.second="hoho";

    std::vector<int> v;
    // for(int i=0; i<40; i++) v.push_back(i%15);

    // for(int i=0; i<40; i++) qDebug()<<v[i];

    sort(v.begin(),v.end());
    //for(int i=0; i<40; i++) qDebug()<<v[i];

    int tmp_zmienna=7;

    qDebug()<<"tmp_zmienna="<<tmp_zmienna;

    mnoz(tmp_zmienna,3);
    qDebug()<<"tmp_zmienna="<<tmp_zmienna;

    printf("asdfasfd");

    qDebug()<<"DONE.";

    return a.exec();
}
```