Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Глазовский государственный педагогический институт имени В.Г. Короленко»

**ОТЧЕТ**

**по учебной практике**

**ПМ 01 Разработка программных модулей программного обеспечения для компьютерных систем**

**Специальность 09.02.03 Программирование в компьютерных системах**

|  |
| --- |
| Выполнил(а): |
| студент(ка) ОП СПО группа 21 |
| Баженов Д.А. |
|  |
| Руководитель практики: старший преподаватель кафедры математики и информатики Касаткин К.А. |

г. Глазов 2020

**Оглавление**

[Техническое задание](#_heading=h.lnxbz9) 2

[Аналоги программного продукта](#_heading=h.2zbgiuw) 6

[Используемые технические и программные средства](#_heading=h.1664s55) 7

[Проектная документация](#_heading=h.34g0dwd) 8

[Руководство пользователя](#_heading=h.32w2z2qj5eum) 9

[Руководство администратора](#_heading=h.kxjhkdazgcr1) 9

[Исходный код программы](#_heading=h.7zefam5gnxyg) 10

[Чек-лист тестирования](#_heading=h.1x0gk37) 17

[Отчет тестирования программы](#_heading=h.2afmg28) 17

[Список источников](#_heading=h.pkwqa1) 18

# **Техническое задание**

к учебной практике специальности 09.02.03

на тему «**Цветовой кодировщик RGB to HSB**»

1. **Цель:** разработка однооконного приложения, в котором происходит перевод единиц цвета системы RGB в HSB.
2. **Основные задачи:**
   1. Аналитический обзор аналогичных программных продуктов.
   2. Изучение методов перевода цвета из RGB в HSB.
   3. Создание приложения.
3. **Используемые программные и технические средства:**
   1. Персональный компьютер стандартной конфигурации.
   2. Операционная система Windows 10 Enterprise.
   3. Среда разработки Qt Creator v4.11.1.
4. **Состав системы:**
   1. Подсистема хранения информации: база данных о товарах, сотрудниках, клиентах, поставщиках
   2. Подсистема авторизации пользователей.
   3. Подсистема «Гарантийный запас товаров».
   4. Подсистема «Оптимальный размер заказа».
   5. Подсистема «Отчетность».
5. **Требования к системе**

**5.1** Использование в системе интуитивно понятного, ненагруженного интерфейса.

**5.2** Решение задачи перевода значений цвета в системе RGB в систему HSB.

1. **Экономическая часть:**
   1. Составление описания проектируемого продукта и направления его использования.
   2. Разработка календарного план-графика работы над проектом.
   3. Расчет сметы затрат на разработку НТР.
   4. Расчет себестоимости и цены программного продукта.
   5. Обоснование экономической эффективности проекта.
2. **Охрана труда:**
   1. Организация оптимального рабочего места с ПЭВМ.
   2. Анализ и устранение вредных и опасных факторов, возникающих при работе с ПЭВМ.
   3. Мероприятия по предотвращению и уменьшению влияния вредных факторов.
      1. Эргономика рабочего места.
      2. Режим труда.
      3. Обеспечение пожарной безопасности.
      4. Обеспечение электробезопасности.
      5. Расчет освещения в помещении.
3. **Состав документации:**
   1. Пояснительная записка.
   2. Руководство пользователя.

Практикант:

студент гр. 21 Д.А. Баженов

Руководитель:

старший преподаватель кафедры информатики, физики, математики К. А. Касаткин

# Аналоги программного продукта

1. Rapidtables [5]

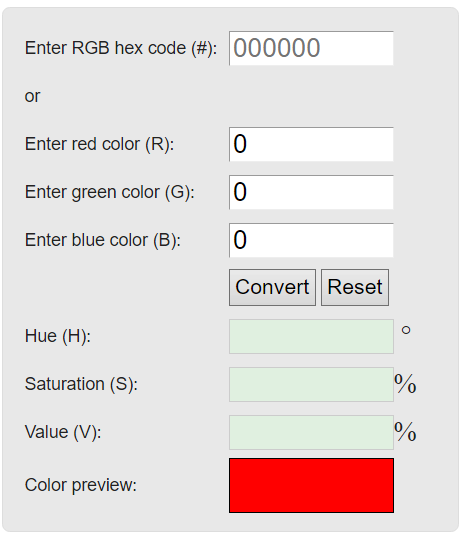


Рисунок 1. Скриншот веб- инструмента Rapidtables

Онлайн-инструмент позволяет переводить цвет из системы RGB в HSB и наоборот, что является достоинством. Однако возможен выход за границы допустимых значений от 0 до 255, считаю это недостатком.

2. Colorscheme [2]

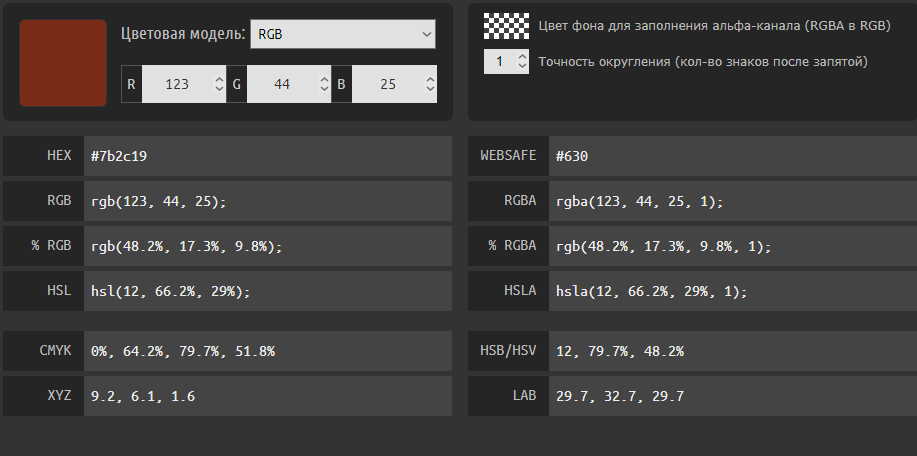


Рисунок 2. Скриншот веб-инструмента Colorscheme

Данный инструмент позволяет переводить цвета в большинство цветовых моделей. В первую очередь, калькулятор служит для удобной работы с современными форматами CSS3-цветов.

# Используемые технические и программные средства

1. Персональный компьютер стандартной конфигурации.

2. Операционная система Windows 10 Enterprise.

3. Среда разработки Qt Creator v4.11.1.

# Проектная документация

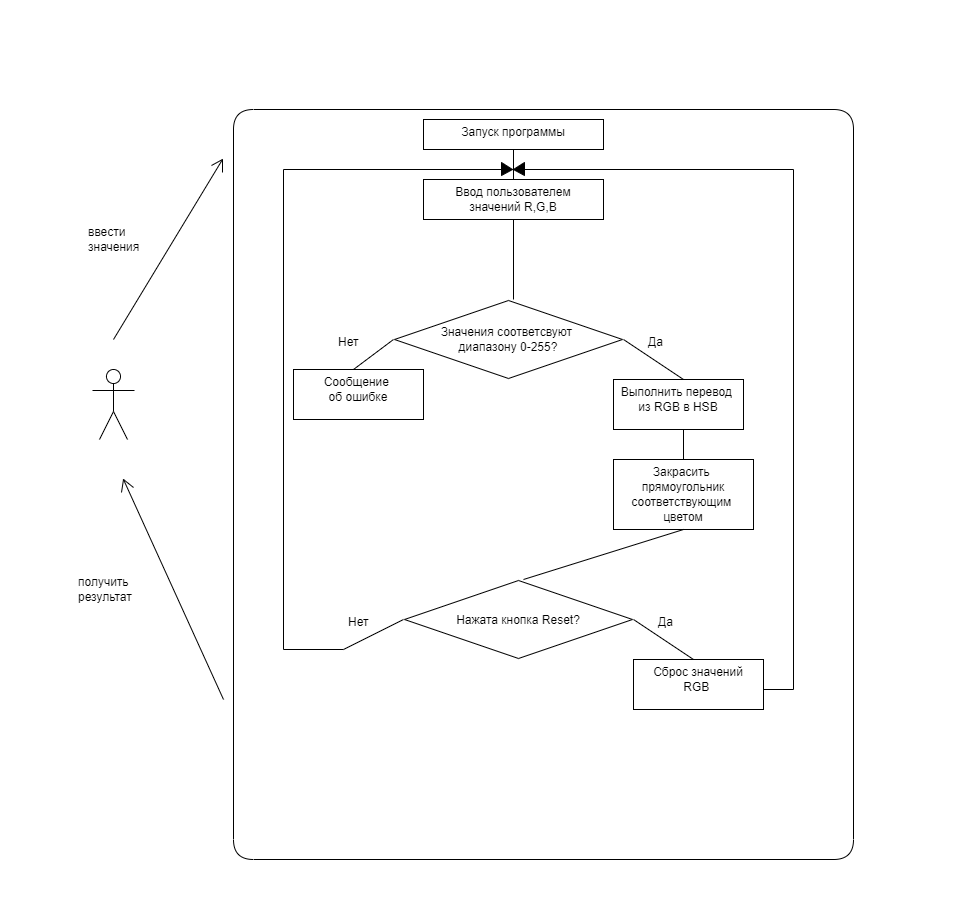


Рисунок 3. Формализация бизнес-процессов предметной области

# Руководство пользователя

Для запуска программы запустите файл RGBtoHSB.exe на своем компьютере. Введите значения параметров RGB в поля ввода и нажмите кнопку Convert. Для сброса параметров нажмите кнопку Reset.

По всем вопросам и предложениям пишите на почту - svz228332@gmail.com.

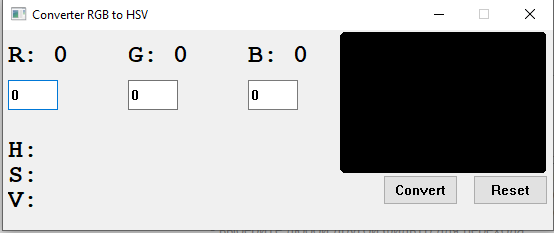


Рисунок 4. Главное окно программы

# 

# Руководство администратора

В случае найденных недочетов в вычислении, откройте файлы исходного кода в среде разработки Qt Creator v4.11.1. Сверьте выводимые программой данные с теми которые выводятся во вкладке “Вывод приложения”. В случае необходимости воспользуйтесь дополнительными операторами qDebug.

# 

# 

# 

# Исходный код программы

QT += core gui

greaterThan(QT\_MAJOR\_VERSION, 4): QT += widgets

CONFIG += c++11

DEFINES += QT\_DEPRECATED\_WARNINGS

SOURCES += \

main.cpp \

mainwindow.cpp

HEADERS += \

mainwindow.h

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QtWidgets>

class MainWindow : public QMainWindow

{

public:

MainWindow();

void paintEvent(QPaintEvent \*event);

void applyBtnClick();

void resetBtnClick();

private:

QLabel \*redLabel, \*greenLabel, \*blueLabel;

QLabel \*labelH, \*labelS, \*labelV;

QLineEdit \*redEdit, \*greenEdit, \*blueEdit;

QPushButton \*applyBtn, \*resetBtn;

};

#endif // MAINWINDOW\_H

#include "mainwindow.h"

#include <QtWidgets>

#include <QList>

#include <QDebug>

float redColor, greenColor,blueColor;

double h,s,v;

double MAX = 0, MIN = 0;

QString strMax, strMin;

QString strRedColor,strGreenColor,strBlueColor;

QString strH,strS,strV;

QColor fillColor;

float var\_R,var\_G,var\_B;

float del;

MainWindow::MainWindow(){

setWindowTitle("Converter RGB to HSV");

setFixedSize(550,200);

strRedColor = QString::number(redColor);

strGreenColor = QString::number(greenColor);

strBlueColor = QString::number(blueColor);

fillColor.setRgb(redColor,greenColor,blueColor,255);

QFont lblFont(QFont("Courier",20,QFont::Bold));

QFont editFont("System", 12, QFont::Normal);

QFont btnFont("Times",16, QFont::Bold);

redLabel = new QLabel(this);

redLabel -> setText("R: "+ strRedColor);

redLabel -> setGeometry(5,5,330,40);

redLabel -> setFont(lblFont);

redEdit = new QLineEdit(this);

redEdit -> setText(strRedColor);

redEdit -> setGeometry(5,50,50,30);

redEdit -> setFont(editFont);

// /// //

greenLabel = new QLabel(this);

greenLabel -> setText("G: "+ strGreenColor);

greenLabel -> setGeometry(125,5,330,40);

greenLabel -> setFont(lblFont);

greenEdit = new QLineEdit(this);

greenEdit -> setText(strGreenColor);

greenEdit -> setGeometry(125,50,50,30);

greenEdit -> setFont(editFont);

// // //

blueLabel = new QLabel(this);

blueLabel -> setText("B: "+ strBlueColor);

blueLabel -> setGeometry(245,5,330,40);

blueLabel -> setFont(lblFont);

blueEdit = new QLineEdit(this);

blueEdit -> setText(strBlueColor);

blueEdit -> setGeometry(245,50,50,30);

blueEdit -> setFont(editFont);

// // //

labelH = new QLabel(this);

labelH -> setGeometry(5,100,330,40);

labelH-> setText("H: "+ strMax);

labelH-> setFont(lblFont);

labelS = new QLabel(this);

labelS -> setGeometry(5,125,330,40);

labelS-> setText("S: "+ strMin);

labelS-> setFont(lblFont);

labelV = new QLabel(this);

labelV -> setGeometry(5,150,330,40);

labelV-> setText("V: "+ strV);

labelV-> setFont(lblFont);

applyBtn = new QPushButton("Convert",this);

applyBtn -> setGeometry(380,145,75,30);

applyBtn -> setFont(editFont);

resetBtn = new QPushButton("Reset",this);

resetBtn -> setGeometry(470,145,75,30);

resetBtn -> setFont(editFont);

connect(applyBtn,&QPushButton::clicked,this,&MainWindow::applyBtnClick);

connect(resetBtn,&QPushButton::clicked,this,&MainWindow::resetBtnClick);

}

void MainWindow::applyBtnClick(){

//первоначальный обработчик значений

strRedColor = redEdit -> text();

redColor = strRedColor.toInt();

redLabel -> setText("R: "+ strRedColor);

strGreenColor = greenEdit -> text();

greenColor = strGreenColor.toInt();

greenLabel -> setText("G: "+ strGreenColor);

strBlueColor = blueEdit -> text();

blueColor = strBlueColor.toInt();

blueLabel -> setText("B: "+ strBlueColor);

// Проверка на переполнение

if (redColor > 256 || greenColor > 256 || blueColor > 256){

QMessageBox::warning(0, "Warning", "RGB parameters out of range");

if (redColor > 256) { redColor = 0; redEdit -> setText("0"); redLabel -> setText("R: 0");}

if (greenColor > 256) { greenColor = 0; greenEdit -> setText("0"); greenLabel -> setText("R: 0");}

if (blueColor > 256) { blueColor = 0; blueEdit -> setText("0"); blueLabel -> setText("R: 0");}

}

else {

// Переменные для вычислений HSV

var\_R = redColor/255;

var\_G = greenColor/255;

var\_B = blueColor/255;

//Нахождение Мин. и Макс. значения

MIN = var\_R < var\_G ? var\_R : var\_G;

MIN = MIN < var\_B ? MIN : var\_B;

MAX = var\_R > var\_G ? var\_R : var\_G;

MAX = MAX > var\_B ? MAX : var\_B;

// разница значений

del = MAX-MIN;

strMax = QString::number(qRound(MAX\*100));

// Вычисление H

if (MAX==MIN){

h = 0;

}

if(MAX==var\_R && var\_G >=var\_B){

h = 60\*((var\_G-var\_B)/del);

}

if(MAX==var\_R && var\_G<var\_B){

h = 60\*((var\_G-var\_B)/del)+360;

}

if(MAX==var\_G){

h = 60\*((var\_B-var\_R)/del)+120;

}

if(MAX==var\_B){

h = 60\*((var\_R-var\_G)/del)+240;

}

// Вычисление S

if (MAX == 0){

s = 0;

}

else {

s = 1-(MIN/MAX);

}

strH = QString::number(qRound(h));

strS = QString::number(qRound(s\*100));

// Вывод вычисленных значений

labelH-> setText("H: "+ strH + "°");

labelS-> setText("S: "+ strS + " %");

labelV-> setText("V: "+ strMax +" %");

fillColor.setRgb(redColor,greenColor,blueColor,255);

update();

}

qDebug() << redColor << greenColor << blueColor;

}

void MainWindow::resetBtnClick(){

redEdit -> setText("0");

greenEdit -> setText("0");

blueEdit -> setText("0");

redColor = 0;

blueColor = 0;

greenColor = 0;

redLabel -> setText("R: 0");

greenLabel -> setText("G: 0");

blueLabel -> setText("B: 0");

labelH-> setText("H: 0°");

labelS-> setText("S: 0 %");

labelV-> setText("V: 0 %");

fillColor.setRgb(0,0,0,255);

update();

qDebug() << redColor << greenColor << blueColor;

}

void MainWindow::paintEvent(QPaintEvent \*){

QPainter painter(this);

painter.setPen(QPen(fillColor,6,Qt::SolidLine,Qt::RoundCap));

painter.setBrush(fillColor);

painter.drawRect(340,5,200,135);

}

#include "mainwindow.h"

#include <QtWidgets>

int main(int argc,char \*argv[])

{

QApplication app(*argc*, argv);

MainWindow window;

window.show();

return app.exec();

}

# Чек-лист тестирования

1. Корректный запуск программы

2. Работоспособность кнопок и полей ввода

3. Правильность перевода цвета

4. Корректность отображения введенного цвета

# Отчет тестирования программы

В ходе тестирования была проверена работоспособность всех функций программы, корректность работы кнопок и полей ввода текста. Также была проверена корректность перевода и отображения цвета.

Результат:

Все функции работают в соответствии с запланированным функционалом.

# Список источников

1. Документы, созданные на практике : <https://drive.google.com/drive/folders/14eNC9-Unpaz3ec2zXrwsIlFwbllfXLMU?usp=sharing>
2. Конвертер Цветов Онлайн. Калькулятор цветов HTML. Генератор цветов CSS3 — ColorScheme.Ru: сайт. — URL: <https://colorscheme.ru/color-converter.html>
3. Qt Documentation: сайт. — URL:<https://doc.qt.io>
4. Online calculator: RGB - HSV and HSV - RGB converter: сайт. — URL:<https://planetcalc.com/5788/>
5. RGB to HSV conversion | color conversion : сайт. — URL:<https://www.rapidtables.com/convert/color/rgb-to-hsv.html>