Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт информационных технологий и анализа данных

Факультет Информационные системы и технологии

Отчёт по лабораторной работе № 3

по дисциплине Технология программирования

Выполнил студент ИСТб-20-3 М. Д. Ярков 27.10.2021

номер группы подпись И. О. Фамилия Дата

Принял Доцент З. А. Бахвалова . .2021

должность подпись И. О. Фамилия Дата

Иркутск 2021

### Постановка задачи

### Создать класс для цвета, заданного в пространстве HSV, а также под каждую характеристику отдельный тип. Реализовать операции:

### Добавление/вычитание красного цвета

### Добавление/вычитание синего цвета

### Добавление/вычитание зеленого цвета

### Добавление/вычитание насыщености

### Добавление/вычитание яркости

### Перевод в RGB

### Словесное описание алгоритма

1. При запуске программы создаётся объект color класса Color, инициализируется интерфейс и производятся изменения значений на форме
2. При нажатии кнопки «Изменить цвета» или нажатии Enter во время изменения красного, зеленого или синего цвета, вызывается метод ChangeRgb:
   1. Производится изменение полей R, G, B объекта color на те, которые были введены:
      1. При изменении R, G или B класс Color автоматически проверяет вхождение изменённого числа в пределы от 0 до 255, и если число недопустимо, появляется ошибка
      2. В случае вхождения в пределы введённое число добавляется во внутреннюю переменную, а также обновляются значения HSV в соответствии с формулой 1
   2. Вызывается метод UpdateLabels:
      1. Производится изменение цвета индикатора на цвет, который содержится в color
      2. Производится изменение надписей интерфейса для отображения значений HSV и RGB
3. При нажатии кнопки «Изменить насыщ. / яркость» или нажатии Enter во время изменения насыщенности или яркости, вызывается метод ChangeHsv:
   1. Производится изменение полей S, V объекта color на те, которые были введены:
      1. При изменении S, V класс Color автоматически проверяет вхождение изменённого числа в пределы от 0 до 100 (для H максимальное значение 360), и если число недопустимо, появляется ошибка
      2. В случае вхождения в пределы введённое число добавляется во внутреннюю переменную, а также обновляются значения RGB в соответствии с формулой 2
   2. Вызывается метод UpdateLabels

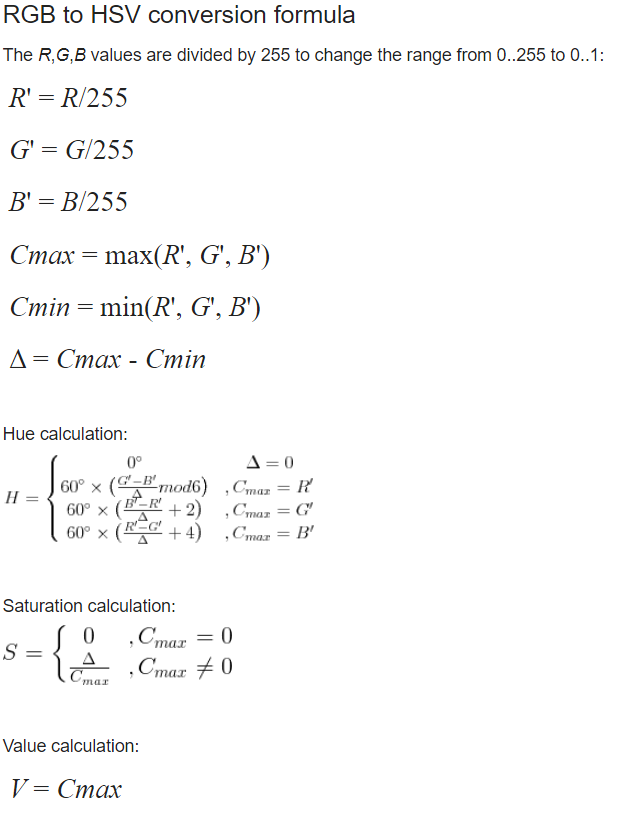


Рисунок 1 – Математическое описание алгоритма перевода RGB в HSV

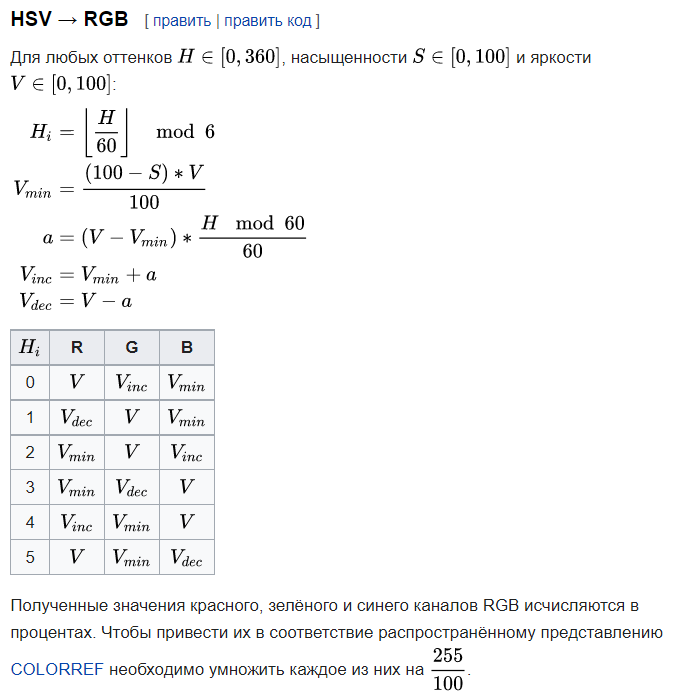


Рисунок 2 – Математическое описание алгоритма перевода HSV в RGB

### Таблица спецификаций

Класс Form1: основной публичный класс, позволяющий взаимодействовать с интерфейсом. Содержит конструктор public Form1() для инициализации интерфейса при первом запуске и методы:

* private void UpdateLabels() – вывод данных в интерфейсе
* private void ChangeRgb(object sender, EventArgs e) - изменение значений RGB при нажатии кнопки в интерфейсе
* private void KeyPressedRgb(object sender, KeyEventArgs e) – изменение значений RGB при нажатии Enter после ввода данных в полях R/G/B
* private void ChangeHsv(object sender, EventArgs e) - изменение значений HSV при нажатии кнопки в интерфейсе
* private void KeyPressedHsv(object sender, KeyEventArgs e) – изменение значений HSV при нажатии Enter после ввода данных в полях S/V

Таблица 1 - поля и переменные класса Form1

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля / переменной | Назначение | Тип | Диапазон |
| **Поля класса** | | | |
| color | взаимодействие с классом | объект readonly Color | - |
| **private void ChangeRgb(object sender, EventArgs e)** | | | |
| sender | не используется | sender | - |
| e | не используется | EventArgs | - |
| **private void KeyPressedRgb(object sender, KeyEventArgs e)** | | | |
| sender | не используется | sender | - |
| e | указывает на нажатую кнопку | EventArgs | - |
| **private void ChangeHsv(object sender, EventArgs e)** | | | |
| sender | не используется | sender | - |
| e | не используется | EventArgs | - |
| **private void KeyPressedHsv(object sender, KeyEventArgs e)** | | | |
| sender | не используется | sender | - |
| e | указывает на нажатую кнопку | EventArgs | - |

Класс Color: публичный класс для работы с цветом. Содержит конструктор public Color() для инициализации в цветовых моделях RGB и HSV, а также методы, геттеры и сеттеры:

* public int H – геттер/сеттер свойства цвета H
* public int S – геттер/сеттер свойства цвета S
* public int V – геттер/сеттер свойства цвета V
* public int R – геттер/сеттер свойства цвета R
* public int G – геттер/сеттер свойства цвета G
* public int B – геттер/сеттер свойства цвета B
* private void \_UpdateRgb() – изменение свойств цвета R, G, B, используется после изменения свойств H, S, V через сеттеры
* private void \_UpdateHsv() – изменение свойств цвета H, S, V, используется после изменения свойств R, G, B через сеттеры

Таблица 2 - поля и переменные класса Color

| Имя поля / переменной | Назначение | Тип | Диапазон |
| --- | --- | --- | --- |
| **Поля класса** | | | |
| \_h | свойство цвета H | private int | от -2147483648 до 2147483647 |
| \_s | свойство цвета S | private int | от -2147483648 до 2147483647 |
| \_v | свойство цвета V | private int | от -2147483648 до 2147483647 |
| \_r | свойство цвета R | private int | от -2147483648 до 2147483647 |
| \_g | свойство цвета G | private int | от -2147483648 до 2147483647 |
| \_b | свойство цвета B | private int | от -2147483648 до 2147483647 |
| **public Color(ColorModels model = ColorModels.HSV, int var1 = 180, int var2 = 50, int var3 = 50)** | | | |
| model | выбор модели цвета | ColorModels | константы HSV и RGB |
| var1 | модификатор H или R | int | от -2147483648 до 2147483647 |
| var2 | модификатор S или G | int | от -2147483648 до 2147483647 |
| var3 | модификатор V или B | int | от -2147483648 до 2147483647 |
| **private void \_UpdateRgb()** | | | |
| vMin | промежуточная переменная для изменения одного из свойств | int | от -2147483648 до 2147483647 |
| a | промежуточная переменная для изменения одного из свойств | int | от -2147483648 до 2147483647 |
| vInc | промежуточная переменная для изменения одного из свойств | int | от -2147483648 до 2147483647 |
| vDec | промежуточная переменная для изменения одного из свойств | int | от -2147483648 до 2147483647 |
| **private void \_UpdateHsv()** | | | |
| r | свойство R, представленное в промежутке между 0 и 1 | double | от (+/-)5.0 x 10-324 до (+/-)1.7 x 10308 |
| g | свойство G, представленное в промежутке между 0 и 1 | double | от (+/-)5.0 x 10-324 до (+/-)1.7 x 10308 |
| b | свойство B, представленное в промежутке между 0 и 1 | double | от (+/-)5.0 x 10-324 до (+/-)1.7 x 10308 |
| min | минимальное из r, g и b | double | от (+/-)5.0 x 10-324 до (+/-)1.7 x 10308 |
| max | максимальное из r, g и b | double | от (+/-)5.0 x 10-324 до (+/-)1.7 x 10308 |

### Таблица тестов

| Номер теста | Что проверяем | Входные данные | Выходные данные |
| --- | --- | --- | --- |
| 1 | Изменение любого цвета в разрешённых промежутках | 30  60  120 | Установлены соответствующие цвета, индикатор перекрашивается в другой цвет, обновлено представление в HSV |
| 2 | Изменение насыщенности/яркости в разрешённых промежутках | 95  60 | Установлена соответствующая насыщенность/яркость, индикатор перекрашивается в другой цвет, обновлено представление в RGB |
| 3 | Попытка установить значение за возможными пределами | -200 | Автоматическое изменение значения поля на ближайшее к введённому разрешённое число |

### Код программы

Код программы находится по ссылке https://github.com/mxkmn/RepoForMyTeacher\_CSharpEdition/tree/main/Lab3Var12

Файл с автотестами:

|  |
| --- |
| using Microsoft.VisualStudio.TestTools.UnitTesting;  using Lab3Var12;  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Lab3Var12.Tests {  [TestClass()]  public class ColorTests {  [TestMethod()]  public void ColorsCreateTest1() {  Assert.ThrowsException<System.ArgumentException>(() => new Color(ColorModels.HSV, 100, 100, 101));  }  [TestMethod()]  public void ColorsCreateTest2() {  Assert.ThrowsException<System.ArgumentException>(() => new Color(ColorModels.HSV, 100, 100, -1));  }  [TestMethod()]  public void ColorsCreateTest3() {  Assert.ThrowsException<System.ArgumentException>(() => new Color(ColorModels.RGB, 100, 100, 256));  }  [TestMethod()]  public void ColorsCreateTest4() {  Assert.ThrowsException<System.ArgumentException>(() => new Color(ColorModels.RGB, -101, 100, 100));  }  [TestMethod()]  public void ColorsCreateTest5() {  Assert.ThrowsException<System.ArgumentException>(() => new Color(ColorModels.HSV, 361, -100, 0));  }  [TestMethod()]  public void ColorsCreateTest6() {  Assert.ThrowsException<System.ArgumentException>(() => new Color(ColorModels.HSV, -1, 100, 100));  }  [TestMethod()]  public void ColorsCreateTest7() {  Color color = new Color();  Assert.AreEqual(180, color.H);  }  [TestMethod()]  public void ColorsCreateTest8() {  Color color = new Color(ColorModels.HSV, 100, 100, 0);  Assert.AreEqual(100, color.S);  }  [TestMethod()]  public void ColorsCreateTest9() {  Color color = new Color(ColorModels.RGB, 100, 255, 0);  Assert.AreEqual(255, color.G);  }  }  } |

### Тесты

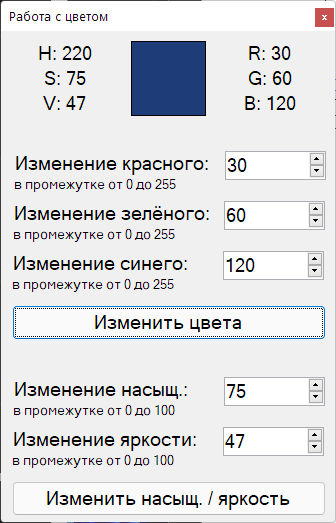


Рисунок 3 – тест 1

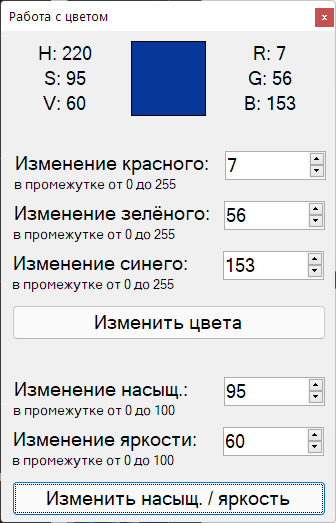


Рисунок 4 – тест 2

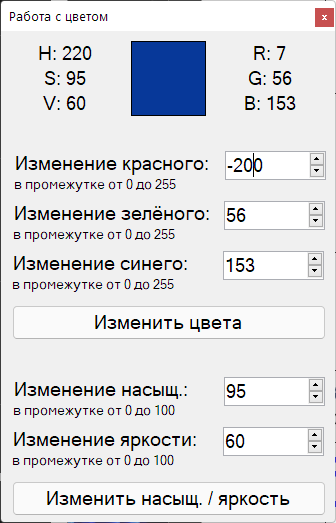


Рисунок 5 – тест 3 (введено число, не входящее в промежуток)

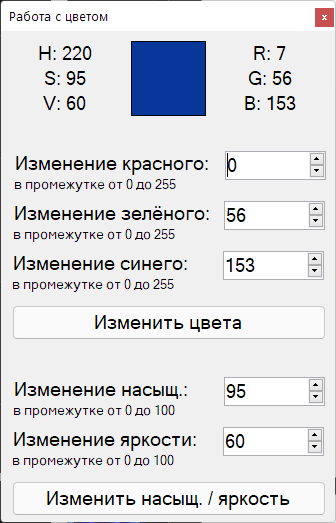


Рисунок 6 – тест 3 (введённое число автоматически изменилось на минимально возможное)