Лабораторная работа №3.

Тема: дополнение графического редактора, созданного в прошлой лабораторной работе.

*Завершающим штрихом для нашего графического редактора будет отдельная форма для выбора цвета пера. Выглядеть она будет следующим образом:*



Для того, чтобы ее добавить, нужно зайти в *проект->добавить форму Windows.*

1. Необходимо разместить все компоненты, как показано на рисунке.

2. Для компонентов *HScrollBar* нужно установить свойство *Minimum* в 0, а *Maximum* в 255. Это будет диапазон изменения оттенков цветов. Так же выставляем *LargeChage* в единицу, это будет величина прокрутки при щелчке на полосе. Точно такие же значения свойств прописываем и для *NumericUpDown*, только вместо *LargeChage* будет свойство *Increment*.

3. Теперь нам нужно связать *HScrollBar* и *NumericUpDown*. Для этого в конструкторе новой формы пропишем следующий код, который свяжет эти 2 компонента через свойство *Tag*:

public ColorsForm(Color **color**)

{

InitializeComponent();

Scroll\_Red.Tag = numeric\_Red;

Scroll\_Green.Tag = numeric\_Green;

Scroll\_Blue.Tag = numeric\_Blue;

numeric\_Red.Tag = Scroll\_Red;

numeric\_Green.Tag = Scroll\_Green;

numeric\_Blue.Tag = Scroll\_Blue;

numeric\_Red.Value = **color**.R;

numeric\_Green.Value = **color**.G;

numeric\_Blue.Value = **color**.B;

}

Для отличия между собой все 6 компонентов были переименованы.

4. Т.к. задача заключается в связывании двух компонентов *HScrollBar* и *NumericUpDown*, то воспользуемся событием *ValueChanged* у обоих компонентов.

Для *HScrollBar*:

private void Scroll\_Red\_ValueChanged(object **sender**, EventArgs **e**)

{

ScrollBar **scrollBar** = (ScrollBar)**sender**;

NumericUpDown **numericUpDown** = (NumericUpDown)**scrollBar**.Tag;

**numericUpDown**.Value = **scrollBar**.Value;

}

Для NumericUpDown:

private void numeric\_Red\_ValueChanged(object **sender**, EventArgs **e**)

{

NumericUpDown **numericUpDown** = (NumericUpDown)**sender**;

ScrollBar **scrollBar** = (ScrollBar)**numericUpDown**.Tag;

**scrollBar**.Value = (int)**numericUpDown**.Value;

}

Аналогично прописывается еще 2 события для оставшихся двух *HScrollBar* и 2 события для двух оставшихся *NumericUpDown*.

5. Теперь нам потребуется одна глобальная для класса переменная *colorResult*:

Color colorResult;

А также отдельная функция (назовем ее *UpdateColor*), которая будет смешивать цвет на основании положения ползунков и закрашивать данным цветом *PictureBox*. Т.к. в данном *PictureBox* мы не будем рисовать, а будем использовать его лишь для определения цвета пера, то инициализировать его

свойство Image не придется, как это делали в предыдущей части лабораторной работы. Для *UpdateColor* не принципиально откуда брать значения – из *HScrollBar* или из *NumericUpDown*, т.к. они теперь связаны между собой. Для примера взят именно *HScrollBar*:

private void UpdateColor()

{

colorResult = Color.FromArgb(Scroll\_Red.Value, Scroll\_Green.Value, Scroll\_Blue.Value);

picResultColor.BackColor = colorResult;

}

Вызывать ее можно либо в *HScrollBar*, либо в NumericUpDown, т.к. опять же не имеет значения, ведь после 4 пункта мы связали эти 2 компонента. Выбирается любой компонент и для всех трех его цветов вызывается функция *UpdateColor()* в любом месте.

*Фича #1*

*В .Net есть готовое решение по выбору цвета. С ним думаю все знакомы, выглядит оно следующим образом:*



*Данное решение называется ColorDialog. Будем вызывать его по кнопке Other Colors. Помимо того, что будет появляться это окно, нужно будет еще привязать RGB выбранного цвета к в HScrollBar и NumericUpDown. Для этого будет служить следующий код:*

private void buttonOther\_Click(object **sender**, EventArgs **e**)

{

ColorDialog **colorDialog** = new ColorDialog();

if (**colorDialog**.ShowDialog() == DialogResult.OK)

{

Scroll\_Red.Value = **colorDialog**.Color.R;

Scroll\_Green.Value = **colorDialog**.Color.G;

Scroll\_Blue.Value = **colorDialog**.Color.B;

colorResult = **colorDialog**.Color;

UpdateColor();

}

}

6. Теперь переходим к связке двух форм. Связывать их будет лишь один параметр, который будет передаваться из формы 2 по нажатию на кнопку ***Ok*** в форму 1 – это созданный цвет пера *colorResult*. Существует много вариантов по передаче данных из одной формы в другую внутри одного проекта, разберем некоторые из них:

**Изменение модификатора доступа** В Form2 установить модификатор доступа для контрола/поля public В любом месте Form1

|  |  |
| --- | --- |
| Form2 f = new Form2();  f.ShowDialog();  **this**.textBox1.Text = f.textBox1.Text; |  |

**Использование открытого свойства/метода. Способ очень похож на первый** В классе Form2 определяем свойство (или метод)

|  |  |
| --- | --- |
| **public string** Data  { |  |
| **get**  **{**  **return textBox1.Text;**  **}**  **}** | | |  |