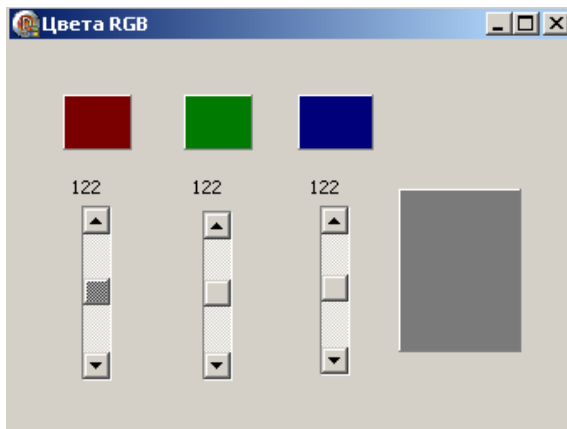


Практическая работа №6, ЦВЕТА В ФОРМАТЕ RGB

Постановка задачи



Создайте программу, с помощью которой пользователь может увидеть в зависимости от значений насыщенности красного, зеленого и синего результирующий цвет (рис.13).

Рис.13

Новым в этой работе являются:

- использование для ввода данных полосы прокрутки **ScrollBar** вкладки палитры компонентов **Standard**,
- компонента панель **Panel** вкладки палитры компонентов **Standard**,
- функция преобразования значений цветовых составляющих – **TColorRef**.

План разработки программы

1. Откройте новый проект.
2. Разместите в форме экземпляры компонентов в соответствии с рис.14.

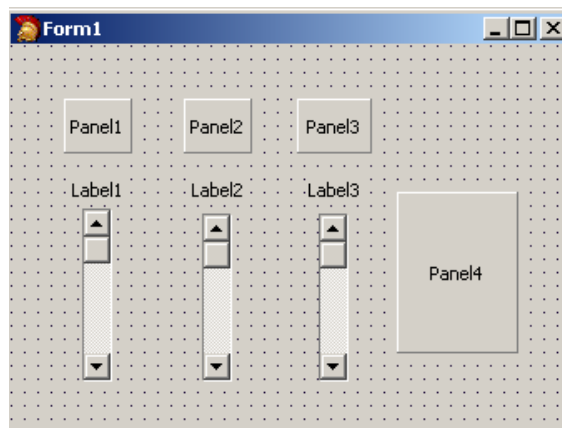


Рис.14

Комментарий

Полоса прокрутки **ScrollBar** может быть горизонтальной (по умолчанию) или вертикальной. Это определяется свойством **Kind**. В нашем случае используется вертикальная полоса прокрутки.

3. Сохраните код программы и проект под именами, например, **Unit6.pas** и **Pr6.dpr**.
4. Выполните следующие действия:

Выделенный объект	Вкладка окна Object Inspector	Имя свойства/ Имя события	Значение/Действие
Panel1	Properties	Name	RedPanel Комментарий Установка имени панели RedPanel , под которым компонент будет известен программе.
		Caption	Удалить название объекта
Label1	Properties	Name	RedLabel Комментарий Установка имени метки RedLabel , под которым компонент будет известен программе.
		Caption	Удалить название объекта
ScrollBar1	Properties	Name	RedBar Комментарий Установка имени полосы прокрутки RedBar , под которым компонент будет известен программе.
		Max	255 Комментарий Максимальный диапазон целых значений – количество градаций компонента RGB.
		Position	122 Комментарий Начальная позиция ползунка – начальное значение.

Аналогично задайте значения для **ScrollBar2, Panel2, Label2**, присвоив им имена **GreenBar, GreenPanel, GreenLabel** и **ScrollBar3, Panel3, Label3**, присвоив им имена **BlueBar, BluePanel, BlueLabel**.

5. Когда на форме много компонентов, ручное выравнивание становится весьма утомительным занятием. Для этого случая в среде Delphi предусмотрены средства автоматизированного выравнивания компонентов.

Выделите компоненты, которые собираетесь выровнять, в нашем случае это **RedLabel (Label1), RedPanel (Panel1), RedBar (ScrollBar1)**. Во всех четырех углах каждого выделенного компонента появятся небольшие квадратики-маркеры. А затем вызовите команду главного меню **Edit/Align**, в результате откроется окно Alignment (рис.15).

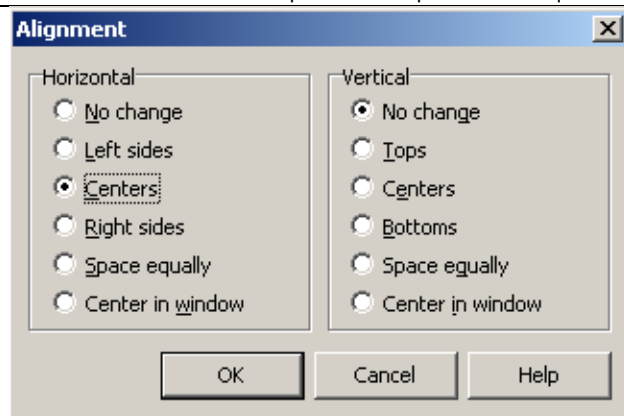


Рис.15

Выберите в списке нужный режим выравнивания и нажмите клавишу «OK».

Повторите эту же операцию для других групп компонент (**GreenBar**, **GreenPanel**, **GreenLabel** и **BlueBar**, **BluePanel**, **BlueLabel**).

6. Выполните следующие действия:

Выделенный объект	Вкладка окна Object Inspector	Имя свойства/Имя события	Значение/Действие
RedBar (см. п.4 - ScrollBar1)	Events	OnChange	<pre>RedPanel.Color:= TColorRef (RGB (RedBar.Position, 0, 0)); RedLabel.Caption:=IntToStr (RedBar.Position); Panel4.Color:= TcolorRef (RGB (RedBar.Position, GreenBar.Position, BlueBar.Position));</pre> <p>Комментарий</p> <p>В зависимости от передвижения ползунка RedBar, будет меняться цвет панели RedPanel, выводится числовое значение кода на месте RedLabel и меняться цвет панели Panel4.</p> <p>Функция RGB(R,G,B) превращает три составляющие цвета из трех отдельных значений в одно целое значение цвета. У этой функции три параметра R – значение красного цвета, G – значение зеленого цвета, B – значение синего цвета.</p> <p>В нашем случае в качестве параметров используются значения соответствующих полос прокрутки ScrollBar.</p> <p>TColorRef – это 32-битовое значение, соответствующее цвету, которое получается с помощью функции RGB.</p>

Аналогично задайте значения для **GreenBar** и **BlueBar**, проследите за правильностью записи параметров в функции **RGB**.

7. Сохраните проект окончательно, запустите и протестируйте его.

Задание для самостоятельного выполнения

1. Усовершенствуйте проект:

- а) сделайте к программе заголовок,
- б) вставьте кнопку выхода из программы.

2. Предусмотрите, чтобы при запуске программы были установлены начальные цвета панелей в зависимости от исходных значений ползунков.

Подсказка. Поместите в событие при создании формы (**OnCreate** для Form1) обработку значений позиции **RedBar**, **GreenBar** и **BlueBar**.

3. Внесите изменения так, чтобы выводились на экран значение кода цвета не только в десятичной системе счисления, но и в шестнадцатеричной системе счисления.

Подсказка. Добавьте объект **Label4**. Поместите в событие по изменению полосы прокрутки **ScrollBar** (RedBar) дополнительную строку, изменяющую свойство нового компонента **Label4**:

```
Label4.Caption:=Format(' %x', [RedBar.Position]);
```

Для перевода значений в шестнадцатеричную систему счисления можно воспользоваться функцией **IntToHex**. Описание этой функции можно найти в **Help**.