# Проектные работы в среде Visual Basic

## «Знакомство с интерфейсом системы программирования Visual Basic»

**Цель:** Познакомиться с интерфейсом системы программирования Visual Basic, рассмотреть основные компоненты среды программирования.

# Задание 1.

Создайте и запустите проект, содержащий форму с размещёнными на ней кнопками и текстовым полем. Щелчок по кнопке Command1 должен менять цвет формы на красный и создавать в текстовом поле надпись «Теперь форма красная»; а щелчок по кнопке Command2 должен менять цвет текста на синий и увеличивать его размер, а надпись в текстовом поле должна быть «Теперь текст синий».

# Технология работы:

- 1. Запустите систему программирования Microsoft Visual Basic.
- 2. В окне приглашения выберете тип проекта Standard EXE и нажмите «Открыть».
- 3. Рассмотрите и запомните названия основных элементов окна программы.



- 4. Выберете на Панели объектов кнопку "TextBox" и растяните текстовое поле на форме.
- 5. Аналогично создайте объект CommandButton (кнопка).
- 6. Двойным щелчком по кнопке откройте Окно программного кода и добавьте следующую команду:

Form1.BackColor = vbRed

- 7. Закройте окно кода и запустите проект кнопкой Start на Панели инструментов. Проверьте работу проекта. Остановите проект кнопкой End.
- 8. Теперь снова откройте Окно кода (дважды щёлкнув по кнопке Command1) и добавьте такую команду:

Text1.Text = "Теперь форма красная"

9. Разместите на форме вторую кнопку Command2 и, открыв для неё Окно кода, допишите следующие команды:

Text1.FontSize = 20

# **Text1.ForeColor = vbBlue**

Text1.Text = "Теперь текст синий"

10. Таким образом в окне кода у вас должна быть такая программа: Project1 - Form1 (Code)



- 11. Теперь в режиме проектирования, закрыв Окно кода, отрегулируйте размеры и расположение объектов и размер самой формы.
- 12. Запустите и проверьте работу проекта. У вас должно получиться примерно так:



13. Сохраните проект командой File – Save Project . ВНИМАНИЕ! Обязательно сохраняйте проекты в отдельно созданной папке (поскольку проект сохраняется в виде нескольких файлов).

## Задание 2.

Создайте и сохраните проект следующего содержания:

- кнопка Command1 делает цвет формы зелёным, размер текста в поле Text1 -16пт;

- кнопка Command2 делает цвет формы жёлтым, а цвет текста в поле Text2 делает красным;

🖷, Form1	
Command1	Text1
Command2	Text2
	Command3

- кнопка Command3 делает цвет формы и цвет текста в обоих полях белыми.

## Задание З.

Создайте проект «Фонтомастер» - программу, которая поможет смастерить красивый текст из имеющихся шаблонов.

Левый ряд кнопок меняет цвет текста в текстовом поле, а правый ряд меняет размер шрифта. Таким образом можно подобрать множество различных комбинаций.

🖷, Form1	
Text1	
[Command1]	Command4
Command2	Command5
Command3	Command6

### «Создание электронного альбома»

**Цель:** Формировать умения работать с формой и управляющими элементами (объект Image (Рисунок)).

### Технология работы:

E	ŀ	F	o	i II	n1																											_		C	נ	)	×
•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	·	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	·	•	•	•	•	•	•	•	•	•	•	•	•	•	·	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	·	•	•	•	•	•	•	•	•	•	•	•	•	•	·	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	·	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
								ĺ.														ĺ.											1				
								Ľ.														ĺ.															
																						1													1	1	1
									•										•	•	•	•	•	•	•	•	•	•	•	•	•	•					

1. Запустить программу VB и создать новый проект. В строке заголовка (Caption) вместо слова Form1 записать Электронный альбом.

2. Создать на форме область, в которой будут показываться рисунки. Для этого выбрать компонент Image (Рисунок) на панели компонентов и протягиванием мыши создать на

форме объект Image1. 🖾 Она должна занимать почти всю форму.

- 3. Для настройки свойств области рисунка, на панели свойств выбрать объект Image1. Найти свойство Stretch(Растяжка), у которого только 2 состояния True или False(Включено или Выключено). Если растяжка включена, то размер рисунка будет подгоняться под размер области просмотра: большой рисунок сожмется, а маленький, наоборот, растянется. Включить значение True.
- 4. Найти свойство BorderStyle(Тип рамки) и выбрать значение 1- Fixed Single (Тонкая фиксированная). Рисунок будет более четко выделяться на фоне окна.
- 5. Найти на панели свойств свойство Picture (Изображение) и щелкнуть на нем. Напротив свойства появится кнопка с многоточием. <u>Picture (None)</u> ... Такие кнопки называется построителями. Щелкнуть на кнопке построителя и откроется диалоговое окно Load

Picture(Загрузить рисунок). Выбирать исходный рисунок, после чего щелкнуть на кнопке Открыть.

- 6. Щелкнуть на панели компонентов правой кнопкой мыши откроется контекстное меню. Выбрать в нем пункт Components(Компоненты)откроется одноименное диалоговое окно. Установить в списке флажок возле пункта Microsoft Common Dialog Control 6.0 (Стандартное диалоговое окно) и щелкнуть на кнопке ОК. В нижней части панели появится дополнительная кнопка Common компонентов Dialog (Стандартное диалоговое окно). 🔤 Щелкнуть на кнопке компонента Common Dialog и протягиванием выберите область окна формы вне рисунка. В этом месте появится значок объекта Common Dialog1. Созданный нами объект не должен быть видимым. Это диалоговое окно, которого нет на экране во время работы программы. Такие объекты называют невизуальными. Он нужен только для настройки свойств невидимого объекта. С помощью данного объекта можно открывать разные диалоговые окна, поэтому у него очень много свойств.
- 7. Выбрать на панели свойств пункт(Custom) (Дополнительные свойства). Щелкнуть на кнопке построителя – откроется диалоговое окно Property Pages(Страницы свойств). В поле Dialog Title(Заголовок окна) ввести текст Выбор изображения. В поле FileName(имя файла) ввести полный путь выбранного рисунка (С:\мои документы\мои рисунки\nature5.bmp). В поле DefaultExt(Стандартное расширение) занесем значение .bmp. В поле Flags(флажки) зададим значение 4096. закроем диалоговое окно щелчком по кнопке ОК. Посмотрите панель свойств и убедитесь в том, что сделанные настройки стали значениями объекта.



Property	Pages			X
Open /	Save As Color F	ont Print H	Help	
Dialog	Litle: выбор изобра	жения	<u>Flags:</u> 4100	
FjleNar	те: с:\мои докум	енты\мои рисун	Defa <u>u</u> ltExt:	.bmp
Init <u>D</u> ir:			<u>M</u> axFileSize:	260
Filte <u>r</u> :			FilterI <u>n</u> dex:	0
	Cancel <u>E</u> rror			
	ОК	Отмена	При <u>м</u> енить	Справка

- 8. Дважды щелкнуть на области изображения в окне формы откроется окно кода.
- 9. Записать программу:



CommonDialog1.ShowOpen

Image1.Picture=LoadPicture(CommonDialog1.FileName) Диалоговое окно для выбора файла отображается вызовом метода ShowOpen. Функция LoadPicture возвращает объект, содержащий выбранный рисунок, а мы присваиваем этот объект свойству объекта Image1.

10. Запустить программу, нажав клавишу F5

## «Простой калькулятор»

**Цель:** Систематизировать знания по теме «Арифметические операции», формировать умения работать в среде программирования с использованием функций преобразования типов данных.

Создайте проект калькулятор, производящий простейшие арифметические действия. Внешний вид калькулятора примерно такой:

💐 Супер Каль	кулятор	
Число1		
	+ - ×	:
Число2		
Результат		

Для выполнения расчёта вы набираете в двух верхних текстовых полях первое и второе число, затем щёлкаете по одной из кнопок и в нижнем текстовом поле получаете результат.

### Технология работы:

- 1. Запустите систему программирования Microsoft Visual Basic, выбрав тип проекта Standard EXE.
- 2. Разместите на форме три текстовых поля и четыре кнопки, выбрав их на Панели объектов.
- 3. Щёлкнув по кнопке Command1 задайте в Окне свойств объекта (Properties) значение свойства Caption «+», подберите шрифт надписи с помощью свойства Font, смените значение свойства Name со стандартного на «Кл\_сложение».
- 4. Аналогичные действия произведите с остальными кнопками (значения свойства Name будут «Кл\_вычитание», «Кл\_умножение» и «Кл\_деление»).
- 5. Измените значения свойства Name для текстовых полей на «Число1», «Число2» и «Результат», а значения свойства Text сотрите.
- 6. Расположите на форме слева от каждого поля надпись с помощью объекта Label (Метка). Поменяйте значения свойства Caption на «Число 1», «Число 2» и «Результат». При необходимости смените шрифт надписи свойством Font.
- 7. Добавьте горизонтальную линию над полем «Результат» с помощью объекта Line (Линия). Толщину и цвет линии можно сменить, поменяв значения свойств BorderWidth и BorderColor.
- 8. Откройте Окно программного кода, дважды щёлкнув по клавише сложения, и добавьте такой оператор:

Peзультат.Text = Val(Число1.Text) + Val(Число2.Text)

- 9. Аналогично запрограммируйте и остальные клавиши (используется тот же оператор, но с другим арифметическим действием). Учтите, что умножение обозначается звёздочкой \*, а деление косой чертой /.
- 10. Запустите проект. Проверьте правильность выполнения команд.

### «Перевод чисел из одной системы счисления в другую»

**Цель:** Систематизировать знания по теме «Функции преобразования типов данных», формировать умения работать в среде программирования с использованием функций преобразования типов данных.

1. Создайте форму по образцу.

🛃 Form1			2.
			На
Десятичное число	77		
Восьмеричное число	115	Перевод в восьмеричное	
Шестнадцатеричное число	4D	Перевод в шестнадцатеричное	

форме создать управляющие элементы:

<u>textbox1</u> – для ввода десятичного числа

<u>textbox2</u>– для вывода преобразованного восьмеричного числа

textbox3 – для вывода преобразованного шестнадцатеричного числа

<u>Command1</u> «Перевод в восьмеричное» – кнопка для событийной процедуры, реализующей преобразование в восьмеричное число в строковой форме <u>Command2</u> «Перевод в шестнадцатеричное» – кнопка для событийной процедуры, реализующей преобразование в шестнадцатеричное число в строковой форме.

- 1. Написать программу.
- 2. Запустить программу на выполнение F5.

### «Использование функций даты и времени»

**Цель:** Систематизировать знания по теме «Функции даты и времени», формировать умения работать в среде программирования с использованием функций даты и времени.

### Технология работы:

- 1. Запустите Visual Basic.
- 2. Создайте интерфейс по образцу.



3. Наберите текст программы Private Sub Form\_Load() dat = Date Label2.Caption = Day(dat) Label3.Caption = Month(dat) Label4.Caption = Year(dat) Label5.Caption = WeekDay(dat, vbMonday) End Sub

Private Sub Command1\_Click() d1 = DateValue(Text1.Text) Label7.Caption = "месяцев" Text2.Text = DateDiff("m", d1, Date) End Sub

```
Private Sub Command2_Click()
d1 = DateValue(Text1.Text)
Label7.Caption = "секунд"
Text2.Text = DateDiff("s", d1, Date)
End Sub
Private Sub Command3_Click()
d1 = DateValue(Text1.Text)
Label7.Caption = "дней"
Text2.Text = DateDiff("d", d1, Date)
End Sub
```

4. Запустите на выполнение программу (F5).

# РАЗДЕЛ 2. ОБЪЕКТНО- ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕVISUALBASIC

#### <u>Линейные программы</u>

### «Вычисление площади треугольника по формуле Геррона»

**Цель:** Систематизировать знания по теме «Оператор присваивания», формировать умения разрабатывать линейные программы в среде программирования с использованием текстовых полей и командных кнопок, а также с помощью функций преобразования типов данных Val и Str.

#### Технология работы:

1. Создайте форму по образцу. Для этого на форму поместите четыре

🖳 Площадь треугольника	
Сторона А	
Сторона В	
Сторона С	
Площадь	Вычислить

метки и для каждой определите значение свойства Техt. Первая метка – Сторона А, вторая –Сторона В, третья –Сторона С, четвертая –Площадь. Затем на форму поместите четыре текстовых поля. Каждому из них дайте имя (определите значение свойства Name). Первое поле имеет имя txtA, второе –txtB, третье –txtC, четвертое –txtS. Теперь на форму поместите кнопку. Свойство Техt этой кнопки имеет значение «Вычислить», а свойство Name – btStart.

2. Для того чтобы связать программный код с кнопкой, дважды щелкните левой кнопкой мыши по кнопке. В открывшемся окне редактора программного кода наберите текст программы. Сначала объявите все необходимые переменные. Для решения задачи нам необходимо знать длины сторон треугольника. Это будут переменные a, b, c. Еще потребуется переменная для хранения полупериметра треугольника. Назовем ее р. Результатом работы программы будет треугольника. площадь Соответствующую переменную назовем s. Все переменные будут иметь тип Single, поэтому для их описания можно использовать один оператор Dim.

Dim a, b, c, p, s As Single

Следующий этап –ввод исходных данных. Для нашей задачи исходными данными являются длины сторон треугольника. Это значения переменных a, b, c. Так как все эти переменные имеют числовой тип, то при вводе их значений из текстовых полей необходимо использовать преобразование Val.

a = Val(txtA.Text)

b = Val(txtB.Text)

c = Val(txtC.Text)

Теперь можно вычислить полупериметр. Для этого сложите значения переменных a, b, c и полученную сумму разделите на 2. Значение арифметического выражения запишите в переменную р.

p = (a + b + c) / 2

Затем вычислите площадь треугольника по формуле Герона. Результат запишите в переменную s.

s = Sqrt(p \* (p - a) \* (p - b) \* (p - c))

Заключительное действие в нашей программе –вывод полученного результата. Так как площадь треугольника является числом, то при его выводе необходимо использовать преобразование Str.

txtS.Text = Str(s)

3. Запустите приложение, нажав клавишу F5.

### «Коробка»

**Цель:** Систематизировать знания по теме «Оператор присваивания», формировать умения разрабатывать линейные программы в среде программирования с использованием с помощью функций ввода и вывода.

### Задание.

Создать программу, которая находила бы площадь основания, объём и площадь полной поверхности картонной коробки по заданным размерам: длине, ширине и высоте. Интерфейс программы должен быть примерно такой:

Программа для определения площади основ объёма и площади полной поверхности коро Поехали!	_ 🗆 ×	_									ļ																																																																																																																							
Поехали!	ания, обки	ан обі	Ba O	oı P	H O	сı К	00	О И	4 F1	и 1	(14 C7	0 C1	1	1	1	и П	и 1	и	И	1 Г)	[	1	I	ł	K	(	(	1	1	1	ł	ł	ł	ł	ł	ł	Į	Į	ł	ł	I	]	1	1	]	1	1	1	1	1	1	1	1	1	[	[	[	I	4 T	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	I	I	[ [	[ [	[	1	]	1	I	Į	1	2				Į	Į	1	1	1	I	1	Į	Į	Į	Į	Į	Į	Į	Į		1	Į	Į	Į	Į	Į	Į	Į	Į	Į	Į	Į	Į	1	1	I	1	1	1	1	1	1
		::		:																												•	•		•	•			•	•																																													•																																								•	•	•			

После нажатия кнопки «Поехали» последовательно появляются окна для ввода длины, ширины и высоты коробки. После заполнения всех данных результат выводится в основном окне программы.

#### Технология работы:

- 1. Разместите на форме надпись и кнопку. Определите необходимые свойства объектов и самой формы.
- 2. В окне кода опишите все необходимые переменные (их должно быть минимум 6 длина, ширина и высота коробки, площадь основания, объём и площадь полной поверхности коробки).
- 3. Для кнопки, расположенной на форме, запишите в окне кода команды, позволяющие задать размеры коробки:

w = InputBox("Введите длину коробки") и т.д.

4. Запишите команды для нахождения результатов задачи.

5. Задайте команды для вывода результатов решения задачи на экран: Print "Площадь основания коробки равна "; So и т.д.

6. Запустите проект. Проверьте правильность выполнения команд. Выясните как можно больше недостатков или несовершенств программы и самостоятельно устраните их.

### «Таблица умножения и сложения»

**Цель:** Систематизировать знания по теме «Оператор присваивания», формировать умения разрабатывать линейные программы в среде программирования с использованием горизонтальной полосы прокрутки.

### Технология работы:

1. Создать форму по образцу.

Таблица умножения и сложения * Исходные данные.	_ 🗆 🗵
Выбери первый множитель (от 0 до 1000	)):
Выбери второй множитель (от 0 до 1000	ı):
Результат-	
500 500	
* 500 + 500	-

- 2. Открыть кодовое окно и написать программу.
- 3. Запустить программу на выполнение F5.

## Разветвляющиеся программы. Условные конструкции

### «Добавление и настройка счетчика»

**Цель:** Систематизировать знания по теме «Операторы ветвления», формировать умения разрабатывать разветвляющие программы в среде программирования с использованием различных компонентов.

### Технология работы:

- 1. Запустите программу VB и создадим новый проект.
- 2. Счетчик это не основной компонент, а дополнительный. Для подключения нового компонента щелкните на панели компонентов правой кнопкой мыши. Выбрать пункт Components(Компоненты)- откроется одноименное диалоговое окно.
- 3. Установите в списке флажок возле пункта Microsoft Windows Common Controls-2 (Стандартные элементы управления Windows-2) и щелкните на кнопке ОК. В нижней части панели компонентов появится дополнительные кнопки.
- 4. Создайте форму. Выберите компонент Label (Надпись) на панели компонентов. Поместите надпись в нижнюю часть формы. Настройка надписи. В поле Caption (Заголовок) введем заглавную букву В. Зададим начальные параметры шрифта. Щелкните на свойстве Font (Шрифт), затем на кнопке настройки шрифта. Откроется диалоговое окно Выбор шрифта. В списке Шрифт выберите вариант Times New Roman, задайте размер шрифта 144 пункта. Щелкните на кнопке ОК.
- 5. Откорректируйте вид надписи на форме, при помощи маркеров. Задайте для свойства Autosize (Автоподбор размера) значение True (Да). Открыть диалоговое окно настройки параметров шрифта и задать размер равным 24 пунктам.
- 6. Выровняйте положение надписи по центру формы. Дайте команду Format- Center in Form- Horizontally (Формат- Разместите по центру формы- По горизонтали). Для свойства Alignment (Выравнивание) значение 2- Center (По центру).
- 7. Создайте текстовое поле. На панели компонентов выберите компонент TextBox (Текстовое поле) и нарисуйте этот объект в верхней части формы. Размер поля выберите по своему усмотрению. Задайте для свойства Text (Текст) значение 24.
- 8. Добавляйте счетчик. На панели компонентов выберите компонент UpDown (Счетчик). Выберите его и нарисуйте счетчик в правой части текстового поля или вплотную к нему. Измените высоту счетчика, чтобы она совпадала с высотой текстового поля, а кнопки со стрелками были легко различимы.
- 9. Настройка счетчика. Объект счетчик имеет два режима работы: автономный и присоединенный. Мы будем использовать присоединенный режим. Выделите счетчик левой кнопкой мыши- на панели свойств откроются свойства этого объекта. Задайте для свойства AutoBuddy (Автоприсоединение) значение True (Да).

Свойство BuddyControl (Присоединенный элемент) тут же получит значение Text1. Это название объекта, описывающего текстовое поле. Щелкните на свойстве BuddyProperty (Присоединенное свойство). Выберите в раскрывающемся списке нужное нам свойство объекта Text1- Text(Teкст). Преобразование введенной строки в число счетчик берет на себя. Задайте для свойстваМах (максимальное значение) значение 144, для свойства Min(минимальное значение)- число 1. Текущее значение счетчика хранится в свойстве Value(значение) запишите сюда число 24.

- 10. Откройте окно кода, дважды щелкните на текстовом поле, но не на счетчике. Система VB создаст процедуру. Она обрабатывает событие Change, наступающее в момент изменения текста в поле ввода. Процедура Text1\_Change вызывается независимо оттого, каким способом был изменен текст в поле ввода. В этой процедуре мы запрограммируем все нужные действия.
- 11.Наберите текст программы.

```
Private Sub _Change()
I=Val (Text1. Text)
If I<1 Then I=1
If I>144 Then I=144
Text1.Text=Ltrim$(Str$ (I))
Label1.Font.Size=I
```

End Sub

Параметры шрифта хранятся в свойстве Font (Шрифт).

Размер шрифта определяет свойство Size. Функция Str\$ преобразует число в текст и поэтому добавляет перед ним пробел, чтобы не сливалось с другими словами предложения. Удалить ведущие пробелы в строке позволяет функция Ltrim\$.

12.Запустите программу, нажав клавишу F5.

# «Добавление и настройка движка и флажка»

**Цель:** Систематизировать знания по теме «Оператор условного перехода», формировать умения разрабатывать разветвляющие программы в среде программирования с использованием компонентов: движка и флажка.

### Технология работы:

Задача: запрограммировать таблицу умножения.

Для двух сомножителей нам потребуется два движка, при этом значение, установленное движком, можно увидеть в поле, которое сделано с помощью объекта Надпись. Значение произведения представлено в виде надписей. Для перехода от таблицы умножения к таблице квадратов и обратно применим специальный флажок. Общий вид окна программы:



1. Запустите программу VB и создадим новый проект.

2. Движки это дополнительные компоненты. Для подключения нового компонента щелкните на панели компонентов правой кнопкой мыши. Выбрать пункт Components(Компоненты)- откроется

одноименное диалоговое окно. Установите в списке флажок возле пункта Microsoft Windows Common Controls-6.0 (Стандартные элементы управления Windows) и щелкните на кнопке ОК. В нижней части панели компонентов появится дополнительные кнопки.

- 3. Щелкните на кнопке Slider (Движок) и нарисуйте движок на форме методом протягивания. С помощью панели свойств измените следующие свойства движков.
  - Міп (минимум) зададим значение 2.
  - Мах (максимум) зададим значение 99.
  - Value (значение) это свойство задает текущее положение бегунка, изменяется автоматически, укажем значение 2.
  - SmallChange(малое изменение) бегунок не обязательно перетаскивать с помощью мыши, его можно двигать курсорными клавишами: Влево, Вправо, Вверх, Вниз.
  - LargeChannge (Большое изменение) бегунок также можно двигать клавишами PAGE UP и PAGE DOWN. При этом значение, заданное движком, изменяется на величину, заданную этим свойством, зададим значение равное 7.
  - Tick Frequency (Частота засечек) это свойство указывает, как плотно будет располагаться засечки на полосе движка. Засечки помогают следить за положением бегунка, укажем число 7, такое же как значение свойства LargeChannge (Большое изменение). В этом случае по щелчку на полосе бегунок будет перепрыгивать от одной засечки к следующей.
  - Для второго движка все настройки задайте точно так же.
- 4. Заключите произведение в рамку. Рамка может иметь свой заголовок и окружать группу других элементов управления. Мы добавляем рамку просто для красоты. Для этого служит компонент Frame (Рамка).Выберите его на панели компонентов и нарисуйте рамку на форме методом протягивания, поместите ее ниже и правее движков.
- 5. Измените свойство Caption (Заголовок) введем слово Произведение. Заголовок рамки всегда находится возле ее левого верхнего угла. Изменить его положение нельзя.
- 6. Выберите компонент Label (Надпись) и нарисуйте надпись на форме методом протягивания. Первую надпись разместите справа от верхнего

движка. Вторую надпись поместите рядом со вторым движком, а третью расположите внутри заготовленной рамки. Чтобы числа выглядели красиво, выровняйте правые границы надписей по одной вертикальной линии.

- 7. Настройка надписи. Значение свойства AutoSize (Автоподбор) для всех трех надписей равно False (нет). Это значение задается по умолчанию. Т.к. наши движки установлены в крайнее левое положение, начальное значение множителей равно2, а начальное значение произведения равно 4. Введем нужные значения в свойство Caption (Заголовок) каждой из надписей.
- 8. Измените, значение свойства Alignment(Выравнивание), задайте значение всех трех надписей- Right Justify(Выравнивание по правому краю).
- 9. Выберите компонент CheckBox (Флажок) на панели компонентов и нарисуйте флажок под движками. Измените свойство Caption (Заголовок)- введите подпись Квадрат. Измените, значение свойства Alignment(Выравнивание), выберете Right Justify значение -(Выравнивание по правому краю).
- 10. Щелкните правой кнопкой мыши на верхнем движке. Выберете в контекстном меню пункт View Code (Показать код) откроется окно кода. Выберете Slider1 в окне кода в раскрывающемся списке слева. В списке справа выберите событие Change (Изменение). Система VB создаст процедуру Slider1\_ Change для обработки этого события. Добавим два оператора:

Label1.Caption=Str\$(Slider1.Value)

Label3.Caption=Str\$(Slider1.Value\*Slider2.Value)

11. Для обработки изменения второго движка надо создать аналогичную процедуру- обработчик. Выберите в окне кода в раскрывающемся списке слева объект Slider2. В списке справа выберите событие Change (Изменение). Добавим два оператора: Label2.Caption=Str\$( Slider2.Value)

Label3.Caption=Str\$(Slider1.Value\*Slider2.Value)

- 12. Программируем влияние флажка. Если флажок установлен, бегунки должны всегда иметь одинаковые положения, т. е. Должны двигаться синхронно. Выберите в окне кода в раскрывающемся списке слева объект Check1 (это и есть флажок). Система VB создаст процедуру обработки щелчка Check1\_Click. Приводим движки к одинаковому виду, сделав значение второго движка таким же, как у первого. Записываем оператор Slider2.Value = Slider1.Value. Событие (Щелчок) происходит как при установке, так и при сбросе флажка.
- 13. Вносим изменения в обработчики события Change (Изменение) для движков. Эти процедуры приведены здесь же в окне кода. Выяснить установлен флажок или сброшен, можно с помощью свойства Value (значение). Если флажок сброшен, это свойство содержит 0, а если установлен- 1. Если флажок установлен, приводим второй движок к

значению первого. Добавляем условный оператор: If Check1.Value = 1 Then

Slider2.Value = Slider1.Value

End If

Такие же операторы надо вставить и во вторую процедуру обработки. Отличие состоит в том, что мы приводим первый движок в соответствии со вторым.

If Check1.Value = 1 Then

Slider1.Value = Slider2.Value

End If

- 14. Запустите программу, нажав клавишу F5.
- 15. Полный текст кода программы: Private Sub Check1\_Click()

Slider2.Value = Slider1.Value

End Sub

Private Sub Slider1\_Change()

Label1.Caption = Str\$(Slider1.Value)

```
Label3.Caption = Str$(Slider1.Value * Slider2.Value)
```

If Check1.Value = 1 Then

Slider2.Value = Slider1.Value

End If

End Sub

Private Sub Slider2\_Change()

Label2.Caption = Str\$(Slider2.Value)

Label3.Caption = Str\$(Slider1.Value \* Slider2.Value)

If Check1.Value = 1 Then

Slider1.Value = Slider2.Value

End If

End Sub

# «Использование условного оператора. Приложение «Касса аэрофлота»

**Цель:** Систематизировать знания по теме «Оператор условного перехода», формировать умения разрабатывать разветвляющие программы в среде программирования с использованием компонентов: командная кнопка, текстовое поле, надпись и их свойствами.

## Технология работы:

- 1. Запустите Visual Basic.
- 2. Создайте интерфейс по образцу.

UserForm1				<b>≥</b>
	КАССА АЗ	РОФЛОТ	<b>A</b>	
РЕЙС	ДАТА	ВРЕМЯ		
ФИО	КОЛИЧЕСТВО	LIEHA		
	ИТОГО НА СУММУ			
1 категория 15%	СКИДКИ НА Б	ИЛЕТЫ 3 катен	: : : : : : : гория 25%	
введите категорию	····	юказать сидку на билет		
показать сумму к ог	плате		очистить	

3. Наберите текст программы Option Explicit Dim кол As Integer, цена As Single, сумма As Single Dim коплате As Single, скидка As Single Private Sub CommandButton1\_Click() кол = Val(TextBox5.Text) цена = Val(TextBox6.Text) сумма = кол \* цена TextBox7.Text = Str(сумма) End Sub

Private Sub CommandButton2\_Click() Dim n As Single If TextBox8.Text = "1" Then скидка = (сумма / 100) \* 15 Else If TextBox8.Text = "2" Then скидка = (сумма / 100) \* 20 Else If TextBox8.Text = "3" Then скидка = (сумма / 100) \* 25 TextBox9.Text = Str(скидка) End Sub

Private Sub CommandButton3\_Click() Dim n As Single коплате = сумма - скидка TextBox10.Text = Str(коплате) End Sub

Private Sub CommandButton4\_Click()

```
TextBox1.Text = "": TextBox2.Text = "": TextBox3.Text = "":
TextBox4.Text = "": TextBox5.Text = "": TextBox6.Text = "":
TextBox7.Text = "": TextBox8.Text = "":
TextBox9.Text = "": TextBox10.Text = ""
```

4. Запустите программу F5.

#### «Три шара»

Цель: Систематизировать знания по теме «Операторы ветвления», формировать умения разрабатывать разветвляющие программы в среде программирования с использованием различных компонентов.

### Технология работы:

1. Создайте форму по образцу.

🖳 Form1		
Количество экспериментов 50	15	•
	20	•
	15	•
Три шара		
i bui malia		

Постановка задачи:

k=

Дано число N – количество вызовов функции, которая возвращает шар одного из трех цветов: красный, синий или желтый. Сколько было шаров каждого цвета?

Пусть целое возвращаемое значение k:

1, если шар красного цвета, 2, если шар синего цвета, 3, если шар желтого цвета

2. Наберите текст программы.

# Private Sub Command1\_Click()

```
n = Val(Text1.Text)
k = 0 : c = 0 : g = 0
FillStyle = 0
FillColor = vbBlue
Form1.Circle (5050, 1000), 450, vbBlue
FillColor = vbRed
```

```
Form1.Circle (5050, 2000), 450, vbRed
FillColor = vbYellow
Form1.Circle (5050, 3000), 450, vbYellow
For i = 1 To n
Randomize
\muBeT= CInt(Int((3 * Rnd()) + 1))
If (\muBeT= 1) Then k = k + 1
If (\muBeT= 2) Then c = c + 1
If (\muBeT= 3) Then g = g + 1
Text2.Text = Str(c)
Text3.Text = Str(k)
Text4.Text = Str(g)
Next i
End Sub
```

#### «Работа с переключателями»

Цель: Систематизировать знания по теме «Оператор условного перехода», формировать умения разрабатывать разветвляющие программы в среде

я Кафе		
Выберите кофе и пирожное		
Кофе	Пирожное	
С Экспрессо	С Картошка	
С Капуччино	С Эклер	
С Американо	С Корзиночка	
Ок		

программирования с использованием переключателей.

**Технология работы: 1.** Создайте интерфейс по образцу.

```
2. Наберите текст программы.
If Option1 Then
Cof = Option1.Caption
elseIf Option2 Then
Cof = Option2.Caption
Else
Cof = Option3.Caption
End IF
If Option4 Then
Pir = Option4.Caption
elseIf Option5 Then
Pir = Option5.Caption
Else
Pir= Option6.Caption
End IF
MsgBox Cof & " и " & Pir, , "Ваш выбор"
```

#### «Сопротивление электрической цепи»

Цель: Систематизировать знания по теме «Оператор условного перехода», формировать умения разрабатывать разветвляющие программы в среде программирования с использованием переключателей, рамки.



Технология работы:

1. Создать форму по образцу.

2. Набрать программный код.

Щелчок на кнопке Вычислить Private Sub Command1\_Click() Dim r1, r2 As Single 'значения сопротивлений R1 и R2 Dim r As Single ' сопротивление цепи

'получение исходных данных r1 = Val(Text1.Text)r2 = Val(Text2.Text)

If (r1 = 0) And (r2 = 0) Then Label4.Caption = "Нужно задать величину хотя бы " + \_

"одного сопротивления."



' Переключатели Последовательное соединение и

- 'Параллельное соединение зависимы, поэтому о типе
- соединения можно судить по состоянию одного из этих
- ' переключателей.

```
If Option1.Value = True Then
   'выбран переключатель Последовательное соединение
   r = r1 + r2
 Else
   'выбран переключатель Параллельное соединение
   r = (r1 * r2) / (r1 + r2)
 End If
 Label4.Caption = "Сопротивление цепи: "
 If r < 1000 Then
   Label4.Caption = Label4.Caption + _
   Format$(r, "0.00") + " OM"
 Else
   r = r / 1000
   Label4.Caption = Label4.Caption + _
   Format$(r, "0.00") + " кОм"
 End If
End Sub
```

## Щелчок на переключателе Последовательное соединение

```
Private Sub Option1_Click()

' пользователь изменил тип соединения,

' очистим поле вывода от предыдущего вычисления

Label4.Caption = ""

End Sub
```

## Щелчок на переключателе Параллельное соединение

Private Sub Option2\_Click() Label4.Caption = ""

End Sub

3. Запустить программу F5.

## «Оператор выбора»

**Цель:** Систематизировать знания по теме «Оператор выбора», формировать умения разрабатывать разветвляющие программы в среде программирования с использованием различных компонентов: графическое и текстовое поля, надпись.

## Технология работы:

1. Создать форму по образцу.



Открыть кодовое окно и написать программу.

Private Sub Command1\_Click() p = "C:\Documents and Settings\Администратор\Рабочий стол \Времена года" m = Val(Text1.Text)If m > 0 And  $m \le 12$  Then Select Case m Case 12, 1, 2 Image1.Picture = LoadPicture(p & "зима.bmp") Label2.ForeColor = vbBlue Label2.Caption = " зима " Form1.BackColor = vbBlue Case 3 To 5 Image1.Picture = LoadPicture(p & "BecHa.bmp") Label2.ForeColor = vbGreen Label2.Caption = " весна " Form1.BackColor = vbGreen Case 6 To 8 Image1.Picture = LoadPicture(p & "лето.bmp") Label2.ForeColor = vbRed Label2.Caption = " лето " Form1.BackColor = vbRed Case 9 To 11 Image1.Picture = LoadPicture(p & "осень.bmp") Label2.ForeColor = vbYellow Label2.Caption = " осень " Form1.BackColor = vbYellow End Select End If End Sub 2. Запустить программу на выполнение F5.

## «Использование таймера»

**Цель:** Систематизировать знания по теме «Оператор выбора», формировать умения разрабатывать разветвляющие программы в среде программирования с использованием таймера.

# Технология работы:

1. На форме создайте интерфейс содержащий кнопку Пуск.

Project1 - Microsoft Visual Basic [design] - [Form1 Elle Edit View Project Format Debug Run Tools Add-Ins Window Help F + + + + + + + + + + + + + + + + + + +	<ol> <li>Прорисуйте в любом месте формы управляющий элемент Timer (таймер).</li> <li>В окне свойств установите для таймера интервал 1000 мс (1с).</li> <li>Наберите текст программы.</li> <li>Dim I as byte Private Sub Command1_Click() Timer1.Enabled=True End Sub Private Sub Form_Load()</li> </ol>
I=0 Timer1.Enabled=False With form1 .FontSize=16 .ScaleMode=3 .CurrentX=100 .CurrentY=10 End With End Sub Private Sub Timer1_Timer() i=i+1 Select case i Case 1 ForeColor=vbred Print"Каждый" CurrentX=100 Case 2	
ForeColor=&H80C0FF Print"Охотник" CurrentX=100 Case 3 ForeColor=vbyellow Print"Желает" CurrentX=100	

Case 4 ForeColor=vbBlue Print"Знать" CurrentX=100 Case 5 ForeColor=vbGreen Print"Где" CurrentX=100 Case 6 ForeColor=vbBlack Print"Сидит" CurrentX=100 Case 7 ForeColor=vbBlue Print"Фазан" CurrentX=100 Timer1.Enabled=False End Select End Sub

5. Запустите программу.

### «Функции даты и времени»

**Цель:** Систематизировать знания по теме «Оператор выбора», формировать умения разрабатывать разветвляющие программы в среде программирования с использованием различных компонентов.

## Технология работы:

1. Создать проект по образцу. В проекте рассмотрены функции даты и времени: текущие дата и время, день недели, день месяца, день года, проверка на високосный год, максимальное и минимальное значение в интервале измерения времени, год, день, разность дат. Для каждой функции создается текстовое поле аргумента функции, текстовое результирующее поле, кнопка с названием функции. Управляющие элементы функций, использующих одинаковые исходные поля аргументов окрашены в один цвет.



2. Написать программу. Запуск F5.