

# ОБУЧАЮЩАЯ ПРОГРАММА



# *Базовый уровень знаний*

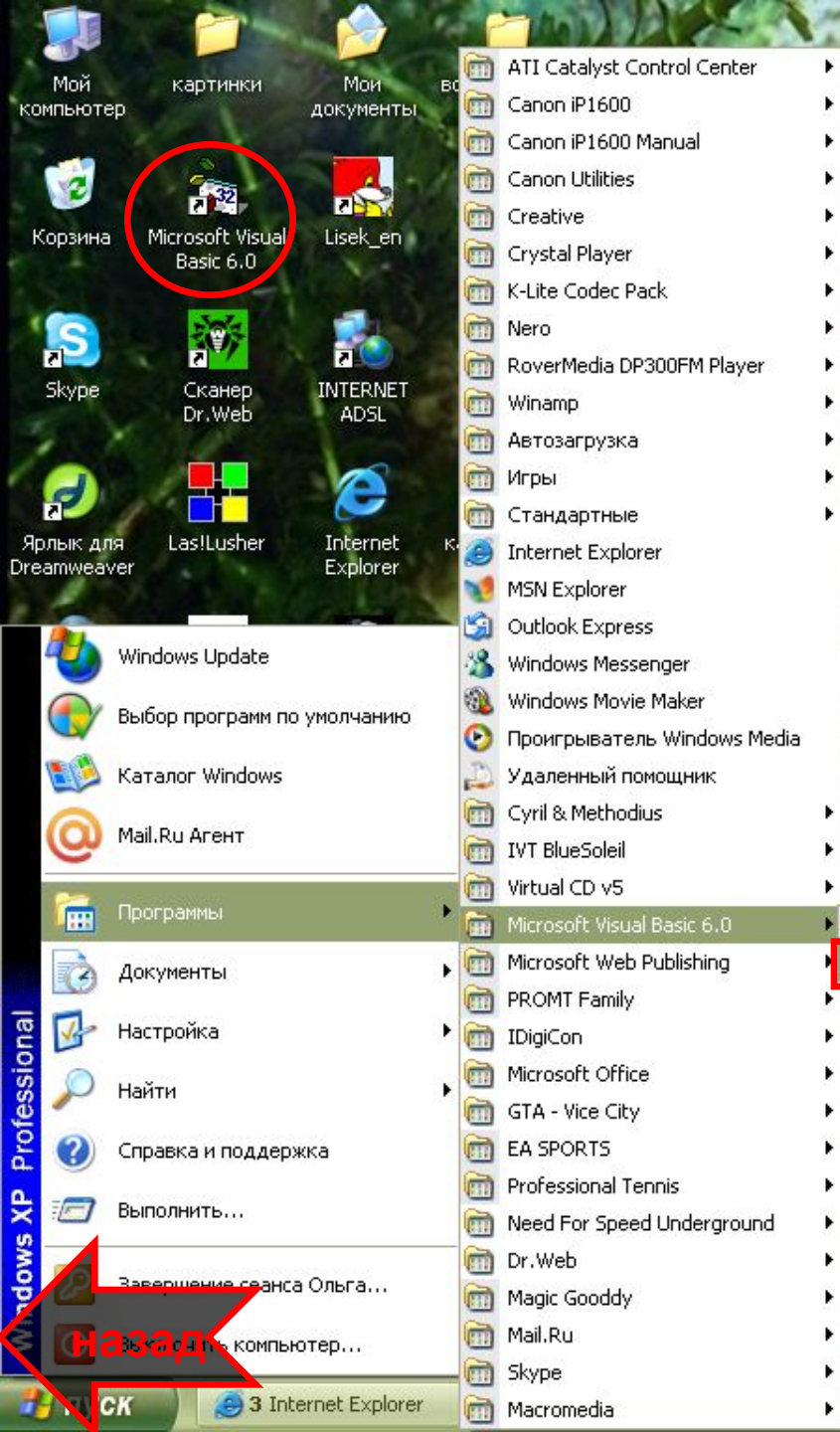
**В работе представлены теоретические сведения и демонстрация работы элементарных программ, созданных с помощью языка объектно-ориентированного программирования **Visual Basic.****



назад

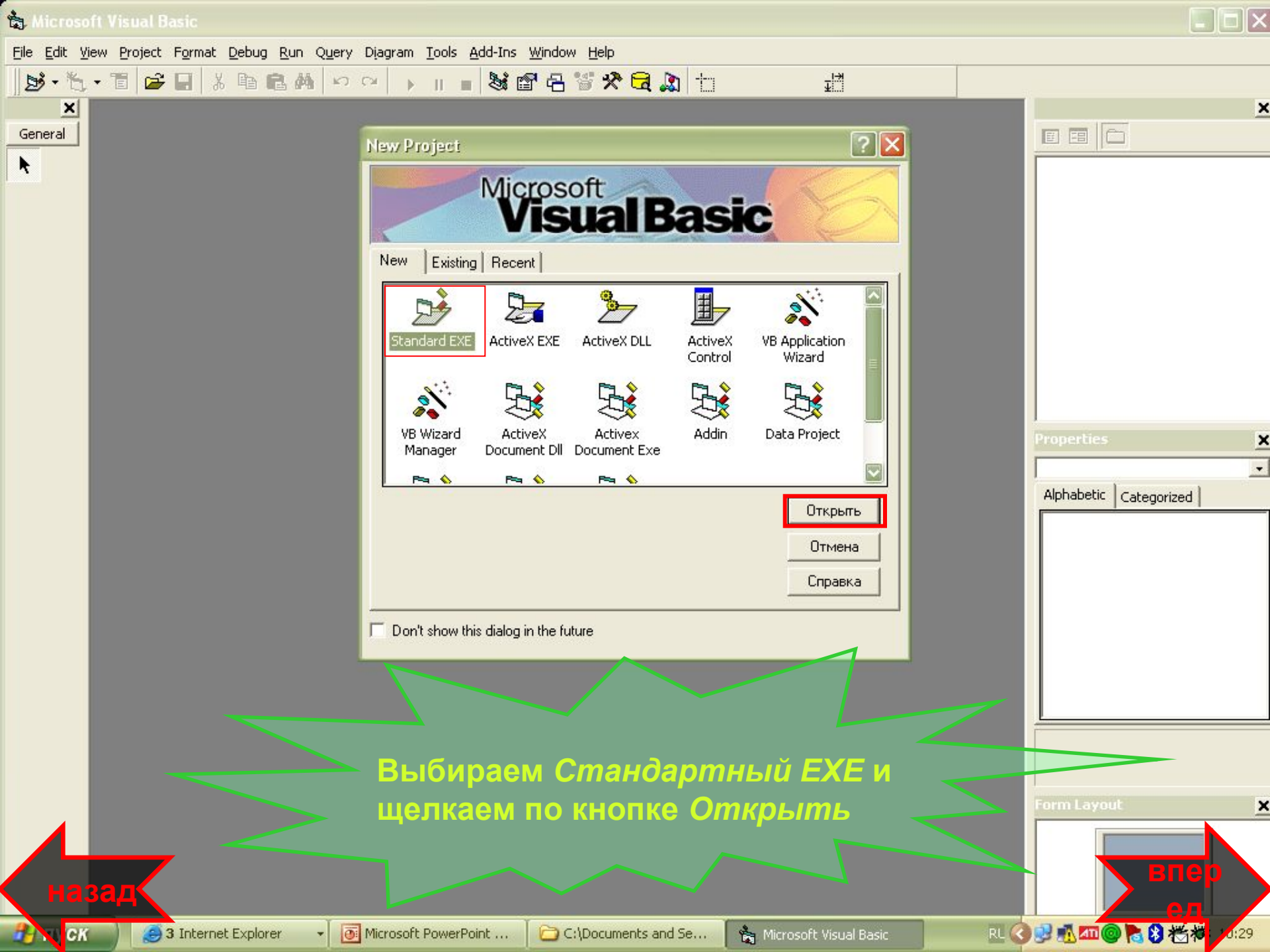


впер  
ед



**Чтобы запустить программу  
можно зайти в нее через меню  
Пуск или щелкнуть по ярлыку  
на рабочем столе**

**впер  
ед**



General

### New Project

Microsoft Visual Basic

New Existing Recent

<b>Standard EXE</b>	ActiveX EXE	ActiveX DLL	ActiveX Control	VB Application Wizard
VB Wizard Manager	ActiveX Document Dll	ActiveX Document Exe	Addin	Data Project

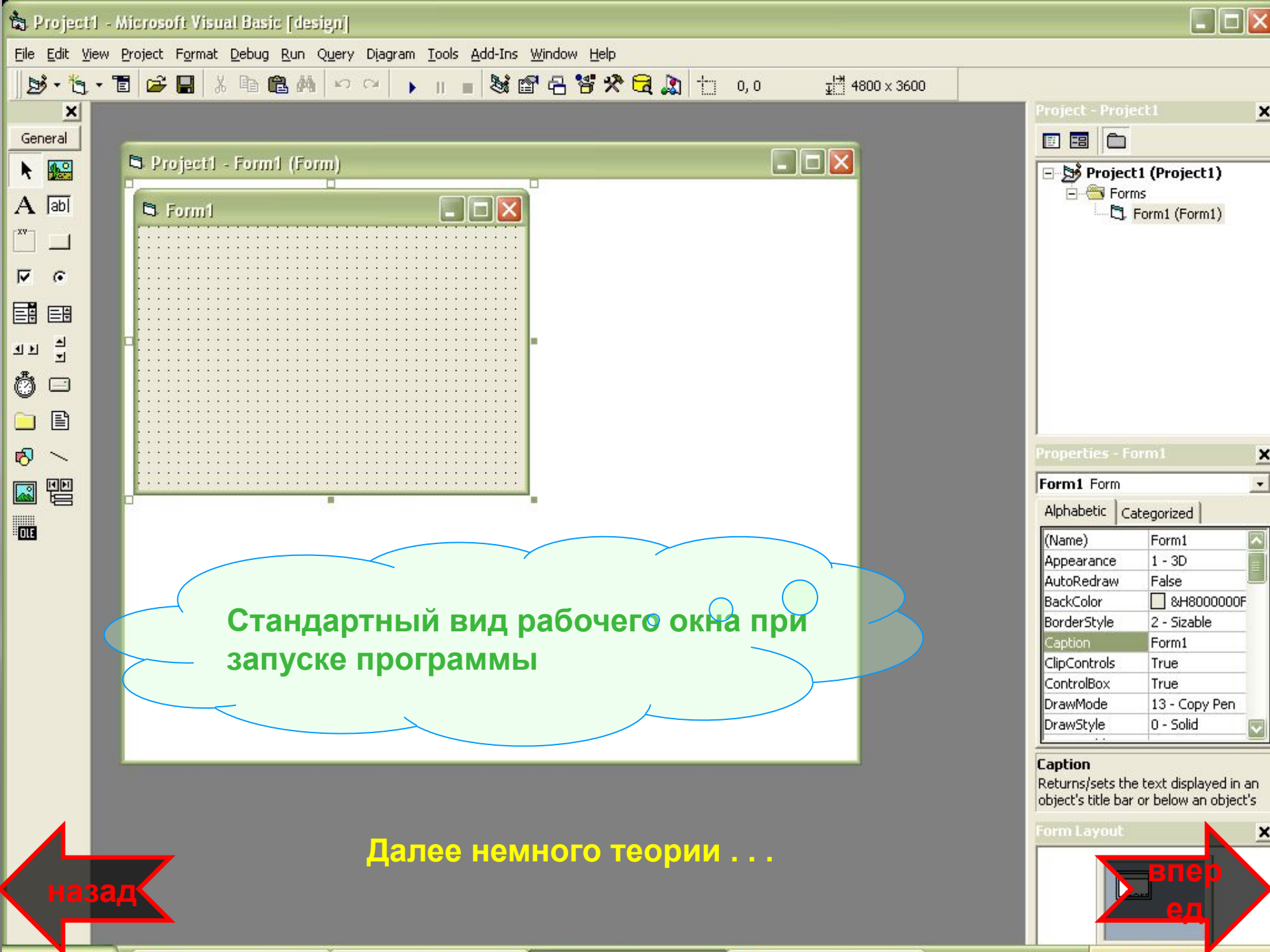
**Открыть**  
Отмена  
Справка

Don't show this dialog in the future

Выбираем *Стандартный EXE* и щелкаем по кнопке *Открыть*

назад

вперед



Стандартный вид рабочего окна при запуске программы

Далее немного теории ...

назад

вперед

# Графический интерфейс

**Графический интерфейс** необходим для реализации интерактивного диалога пользователя с работающим приложением. Основой для создания графического интерфейса разрабатываемого приложения является форма.

**Форма** – это объект, представляющий собой окно на экране, в котором размещают управляющие элементы.


Классы управляющих элементов имеют различное назначение в графическом интерфейсе пользователя.

## Окно конструктор форм

- 1) Открывающееся меню
- 2) Панели инструментов
- 3) Форма
- 4) Окно проводника
- 5) Окно свойств
- 6) Окно расположения формы
- 7) Окно программного кода (вызывается двойным щелчком по форме)



назад

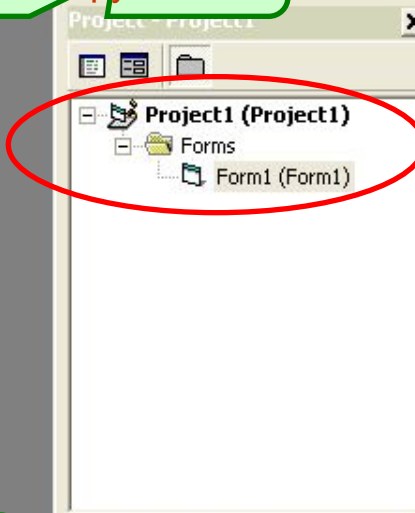


впер  
ед

Линейка  
ИНСТРУМЕНТОВ



Окно проводник  
проекта показывает  
из чего состоит наш  
проект



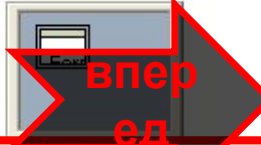
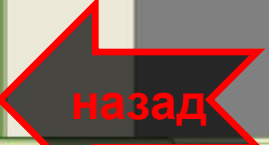
Окно свойств  
выделенного объекта

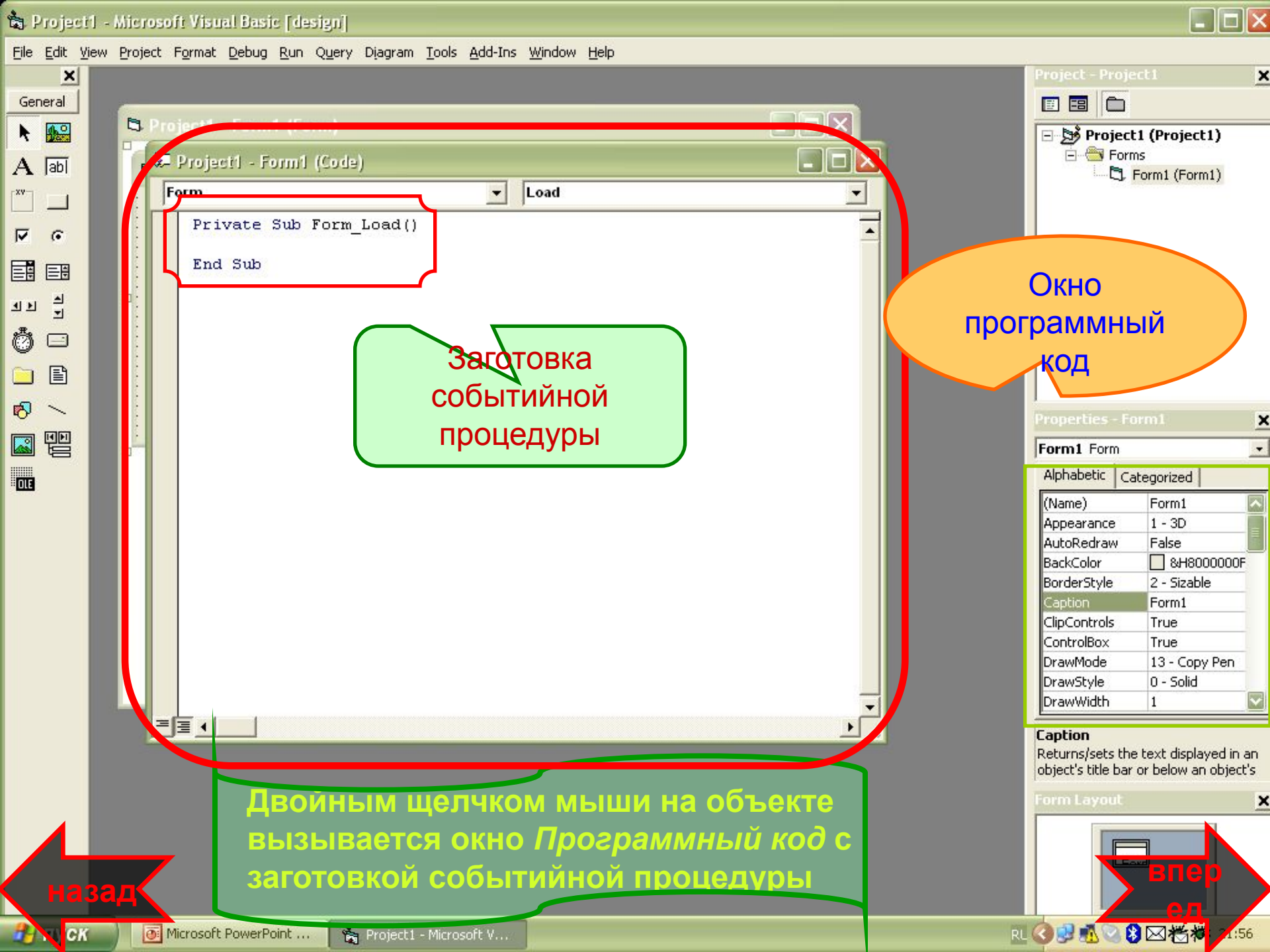


Форма – поле,  
куда помещаются  
управляющие  
элементы

Панель  
инструментов  
ToolBox содержит  
основные  
управляющие  
элементы, которые  
можно вставлять на  
форму

Окно расположения  
формы





```
Private Sub Form_Load()  
  
End Sub
```

Заготовка  
событийной  
процедуры

Окно  
программный  
код

Двойным щелчком мыши на объекте  
вызывается окно Программный код с  
заготовкой событийной процедуры

назад

вперед

Project - Project1

Project1 (Project1)  
Forms  
Form1 (Form1)

Properties - Form1

Form1 Form	
Alphabetic	Categorized
(Name)	Form1
Appearance	1 - 3D
AutoRedraw	False
BackColor	<input type="checkbox"/> &H8000000F
BorderStyle	2 - Sizable
Caption	Form1
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen
DrawStyle	0 - Solid
DrawWidth	1

**Caption**  
Returns/sets the text displayed in an object's title bar or below an object's

Form Layout



# Инструменты панели ToolBox

НАЗВАНИЕ ИНСТРУМЕНТА	ВЫПОЛНЯЕМОЕ ДЕЙСТВИЕ
Pointer	Выделяет объект
PictureBox	Выделяет область для отображения графики
Label	Создает надпись
TextBox	Создает текстовое поле
Frame	Рисует рамку для выделения группы объектов
CommandButton	Создает кнопку
CheckBox	Создает флажок проверки
OptionButton	Создает переключатель
ComboBox	Создает поле со списком
ListBox	Создает список с полосой прокрутки
HScrollBar	Горизонтальная полоса прокрутки
VScrollBar	Вертикальная полоса прокрутки
Timer	Часы
DriveListBox	Список устройств, которые есть на компьютере
DirListBox	Список папок
FileListBox	Список файлов из текущей папки
Shape	Рисует фигуру
Line	Рисует линию
Image	Изображение
OLE	Вставка объекта OLE



назад



вперед

# Некоторые основные свойства объектов

Список свойств упорядочен по алфавиту или по категориям.

Категории свойств:

Appearance – свойства внешнего вида объекта

Behavior – поведение объекта

DDE – обмен данными между приложениями

Misc – смесь

Position – положение и размер

Scale – масштаб

Некоторые свойства основных объектов:

Для **CommandButton**:

(Name) - имя кнопки в программном коде

Caption - надпись на кнопке

Font - изменение параметров шрифта надписи на кнопке

Для **TextBox**:

(Name) - имя текстового поля в программном коде

Text - ввод текста

BackColor - цвет фона текстового поля


Font - изменение параметров шрифта надписи

ForeColor - цвет текста

Alignment – выравнивание текста



назад



впер  
ед

## Для Label:

Name – имя в программном коде

BackColor - цвет фона поля

Caption - надпись в поле

ForeColor - цвет текста надписи

Alignment – выравнивание текста

## Для Form:

Name – имя в программном коде

BackColor - цвет фона формы

Picture – вставка картинки из файла для фона формы



назад



впер  
ед

# Типы переменных

Тип переменной	Возможные значения	Объем занимаемой памяти	Приставка к имени
Byte	Целые числа от 0 до 225	1 байт	byt
Integer	Целые числа от -32768 до 32768	2 байта	int
Long	Целые числа от -2147483648 до 2147483647	4 байта	lng
Single	Десятичные числа одинарной точности $[-1.4 \cdot 10^{-45} - 3.4 \cdot 10^{38}]$	4 байта	sng
Double	Десятичные числа двойной точности $[-5.0 \cdot 10^{-324} - 1.7 \cdot 10^{308}]$	8 байтов	dbl
Boolean	Логическое значение True или False	2 байта	bln
String	Строка символов	1 байт на каждый символ	str
Currency	Число в денежном формате	8 байтов	cur
Date	Дата от 1 января 100 г. до 31 декабря 9999 г.	8 байтов	dtm
Object	Ссылки на любой объект	4 байта	obj
Variant	Любые значения	>=16 байтов	vnt

← назад

→ вперед

# Функции в языке *Visual Basic*

## Функции преобразования типов данных:

**Val** преобразовывает строковое значение в числовое.

**Str, Hex, Oct** преобразовывает десятичные числа в десятичные, восьмеричные и шестнадцатеричные числа в строковой форме.

**Asc** преобразовывает строку в числовой код первого символа.

**Chr** преобразовывает числовой код в соответствующий ему символ.

## Математические функции:

**Sin()** функция синуса

**Cos()** функция косинуса

**Tan()** функция тангенса

**Atn()** функция арктангенса


**Sqr()** квадратный корень

**Exp()** показательная функция

**Rnd()** получение случайного числа



назад



впер  
ед

# Создадим проект «Обычный калькулятор»

Работа над проектом начинается с создания графического интерфейса будущего приложения. В окне *Конструктор форм* на форму помещаются управляющие элементы. Разместим на форме три текстовых поля и пять кнопок. Создадим отделяющие линии.

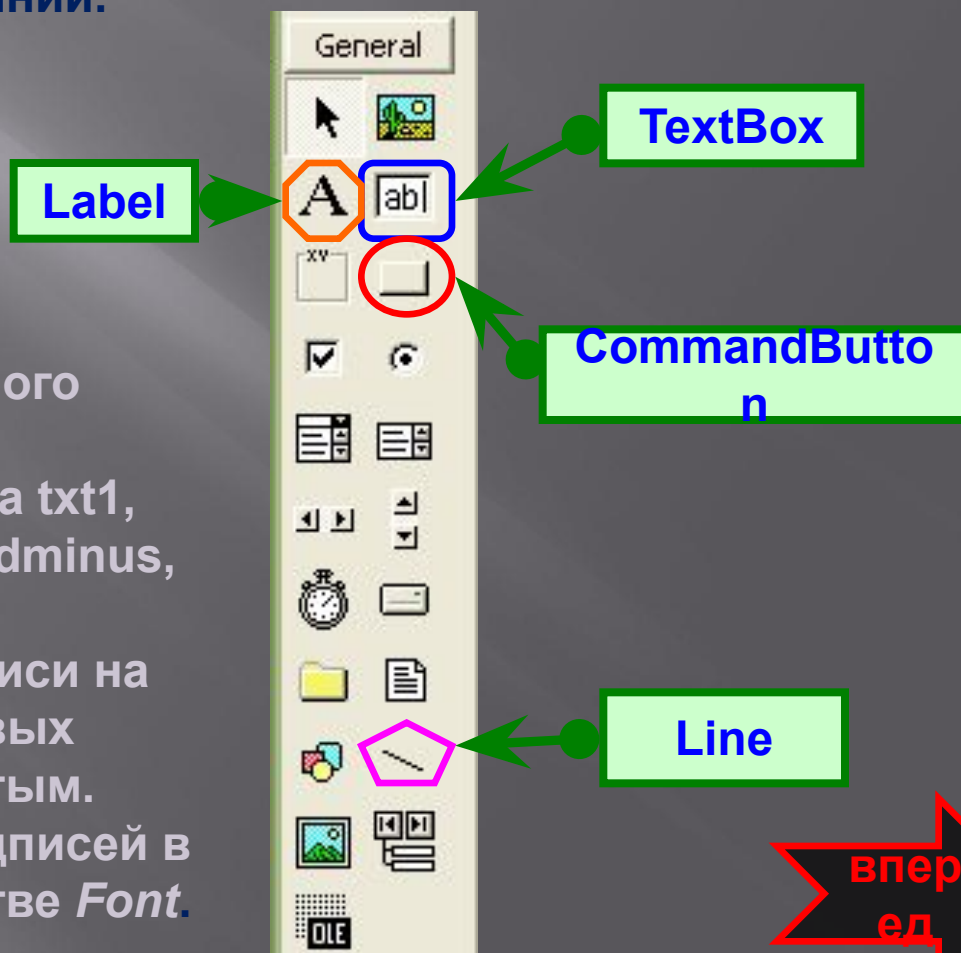
С этой панели инструментов нам необходимо поместить на форму кнопки `CommandButton`, текстовые поля `TextBox`, надписи `Label` и линии `Line`.

Назовем форму «Создание обычного калькулятора».

Присвоим текстовым полям имена `txt1`, `txt2`, `txt3`, а кнопкам – `cmdplus`, `cmdminus`, `cmdumn`, `cmddelen`.

В свойстве *Caption* изменим надписи на кнопках и заголовках. Для текстовых полей свойство *Text* оставим пустым.

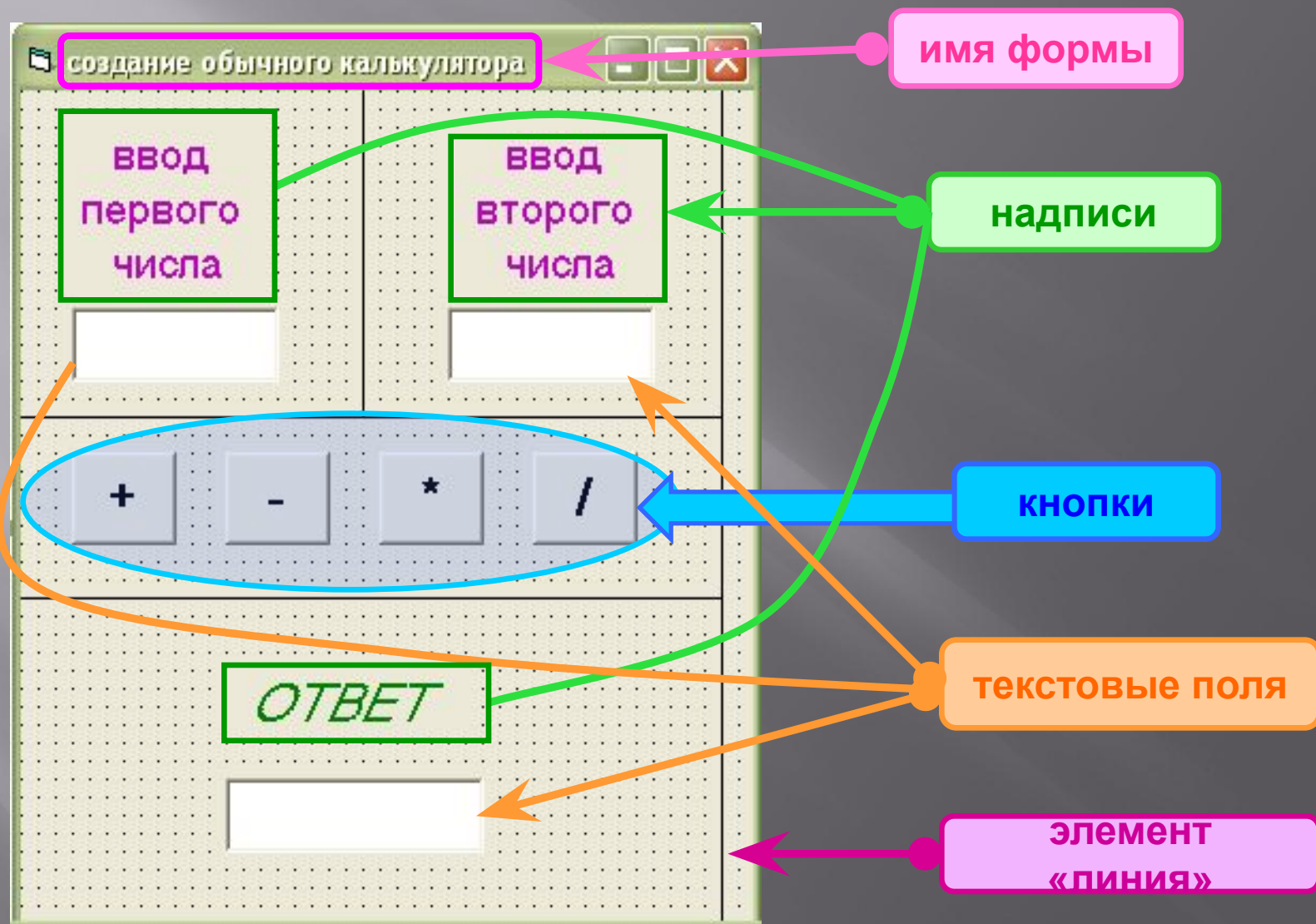
Изменим шрифт и цвет надписей в свойстве *Font*.



назад

впер  
ед

Получим следующий вид графического интерфейса:



назад

вперед

Чтобы программа работала необходимо написать программный код. Событийные процедуры должны изменять значения текстовых полей так, чтобы мы получили сумму числовых значений. Для преобразования строковых значений, вводимых в текстовые поля, в десятичные числа, воспользуемся функцией Val.

Код событийной процедуры будет следующим:

**Список объектов**

**Выполняемое событие**

**Имя объекта, для которого пишется событийная процедура**

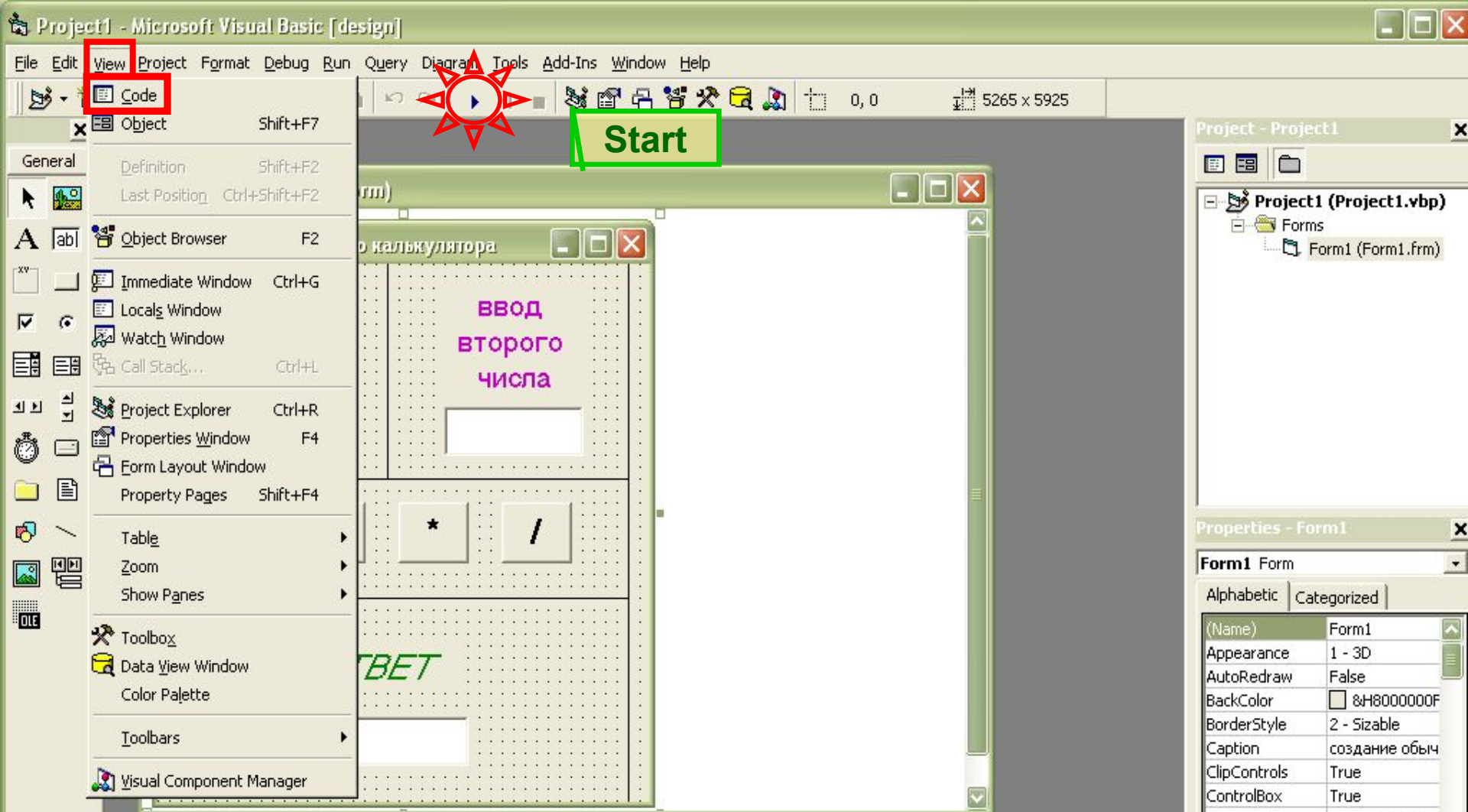
**Список событий для объектов**

```
Private Sub cmddelen_Click()  
txt3.Text = Val(txt1.Text) / Val(txt2.Text)  
End Sub  
  
Private Sub cmdminus_Click()  
txt3.Text = Val(txt1.Text) - Val(txt2.Text)  
End Sub  
  
Private Sub cmdplus_Click()  
txt3.Text = Val(txt1.Text) + Val(txt2.Text)  
End Sub  
  
Private Sub cmdumn_Click()  
txt3.Text = Val(txt1.Text) * Val(txt2.Text)  
End Sub
```

назад

впер  
ед





Теперь запустим программу на выполнение. Это можно сделать несколькими способами:

- 1) Зайти «Вид (View) → Код (Code)»
- 2) Нажать клавишу F5
- 3) Нажать «Старт (Start)» на панели инструментов

← назад

→ вперед



**ВВОДИМ числа  
с клавиатуры**

**выбираем  
действие  
(например  
сложение)  
и нажимаем  
на кнопку**

**получаем  
автоматически  
ответ**

**назад**

**впер  
ед**

# Графические возможности языка Visual Basic



На формах или в графических окнах можно рисовать различные графические примитивы с использованием графических методов:

`Scale (x1, y1) – (x2, y2)` задает систему координат и масштаб для формы или графического окна.  $x_1, y_1$  – координаты верхнего левого угла объекта,  $x_2, y_2$  – координаты правого нижнего угла объекта.

`Pset (x, y)` рисование точки.  $x, y$  – координаты точки.

`Line (x1, y1) – (x2, y2)` рисование линии.  $x_1, y_1$  и  $x_2, y_2$  – координаты концов линии.

`Line (x1, y1) – (x2, y2), BF` рисование прямоугольника.  $x_1, y_1$  и  $x_2, y_2$  – координаты левого верхнего и правого нижнего углов прямоугольника

`Circle (x, y), radius` рисование окружности.  $x, y$  – координаты центра окружности, `radius` – радиус окружности.



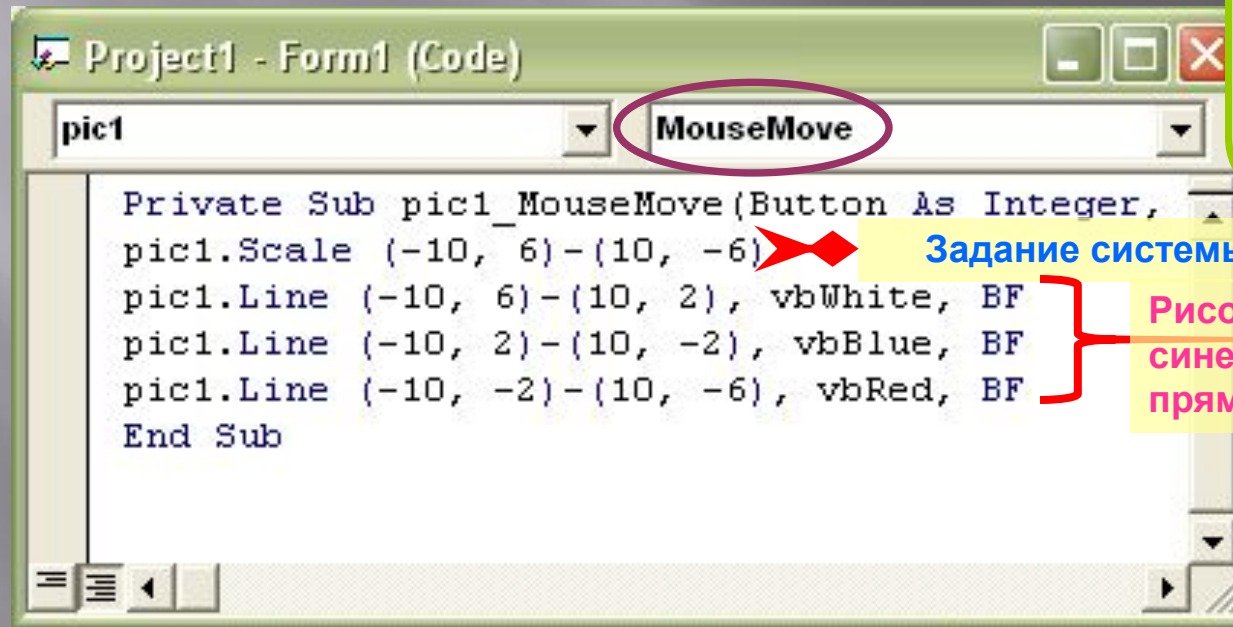
назад

впер  
ед

Для демонстрации графических возможностей Visual Basic создадим программу рисования флага.

Создадим на форме объект PictureBox и зададим имя pic1.

А в окне программного кода надо набрать следующее:



```
Project1 - Form1 (Code)
pic1
MouseMove
Private Sub pic1_MouseMove(Button As Integer,
pic1.Scale (-10, 6)-(10, -6)
pic1.Line (-10, 6)-(10, 2), vbWhite, BF
pic1.Line (-10, 2)-(10, -2), vbBlue, BF
pic1.Line (-10, -2)-(10, -6), vbRed, BF
End Sub
```

Изменим событие после которого произойдет выполнение программы

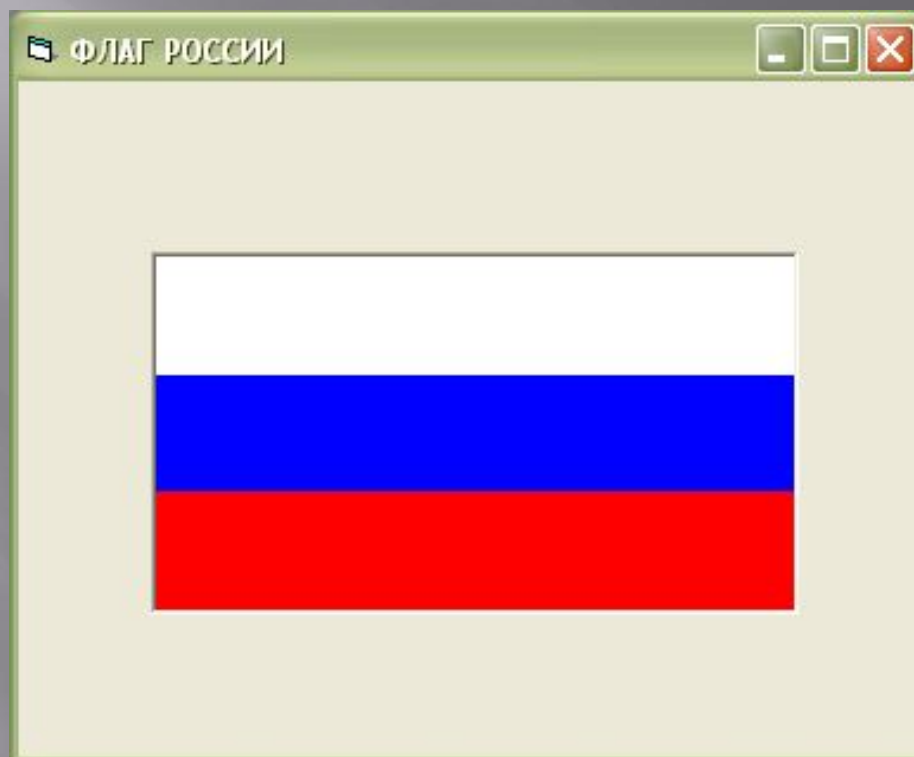
Задание системы координат

Рисование белого, синего и красного прямоугольника



Запустим программу на выполнение...

Проведя мышкой по полю для изображения, мы увидим такую картинку:



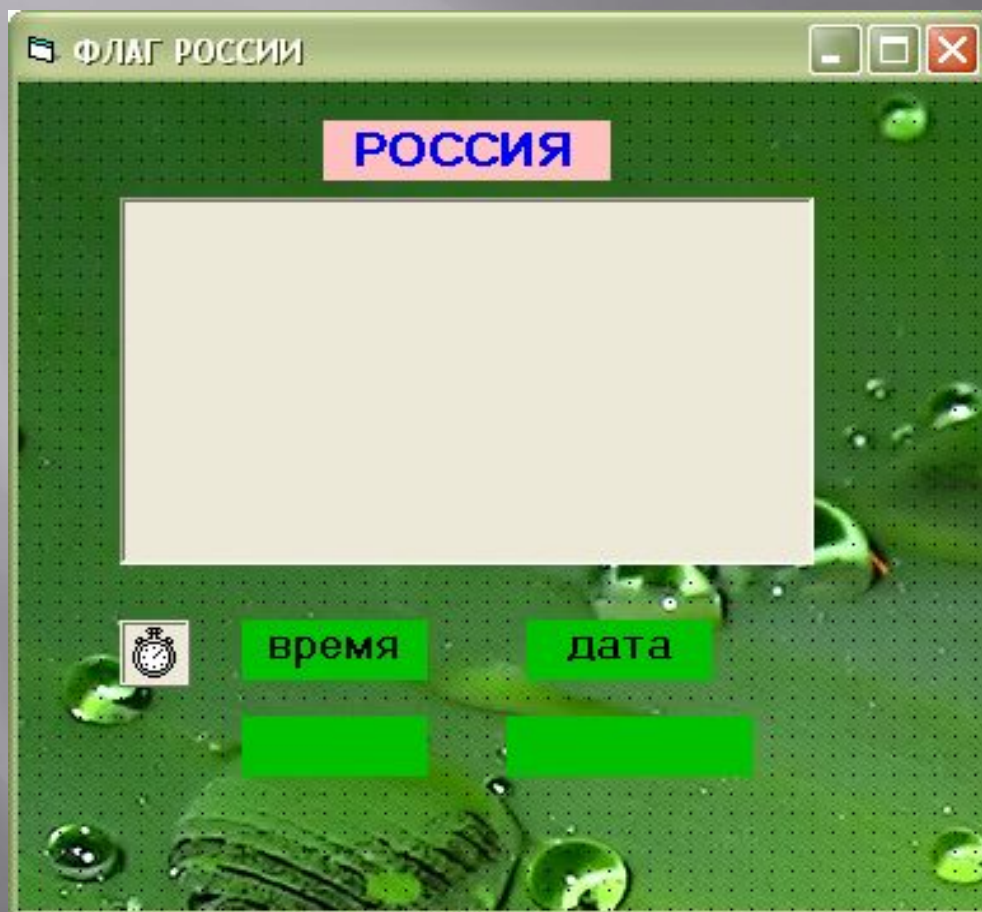
Программу можно дополнить, вставкой автоматического показа времени и даты.



назад

впер  
ед

Дополним интерфейс 5 объектами Label и объектом Timer. Также можно вставить картинку на форму и наш интерфейс будет выглядеть примерно так:



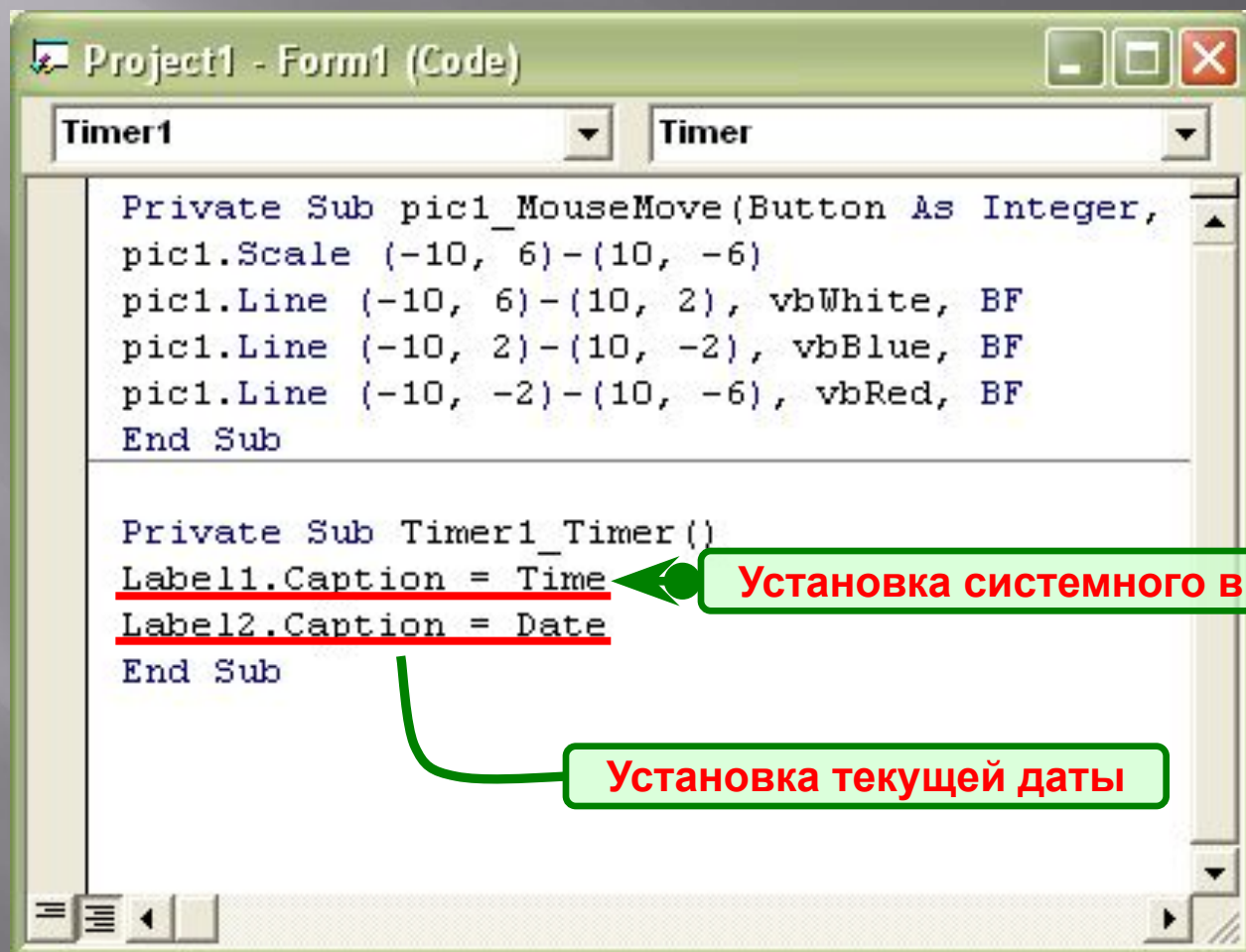
Для того чтобы часы работали, необходимо поменять некоторые свойства объекта Timer:

(Name)	Timer1
Enabled	True
Index	
Interval	1000
Left	720
Tag	333
Top	1440

← назад

→ вперед

Запишем программу запуска даты и времени. Код прописывается под объектом Timer.



```
Project1 - Form1 (Code)
Timer1
Timer

Private Sub pic1_MouseMove(Button As Integer,
pic1.Scale (-10, 6)-(10, -6)
pic1.Line (-10, 6)-(10, 2), vbWhite, BF
pic1.Line (-10, 2)-(10, -2), vbBlue, BF
pic1.Line (-10, -2)-(10, -6), vbRed, BF
End Sub

Private Sub Timer1_Timer()
Label1.Caption = Time
Label2.Caption = Date
End Sub
```

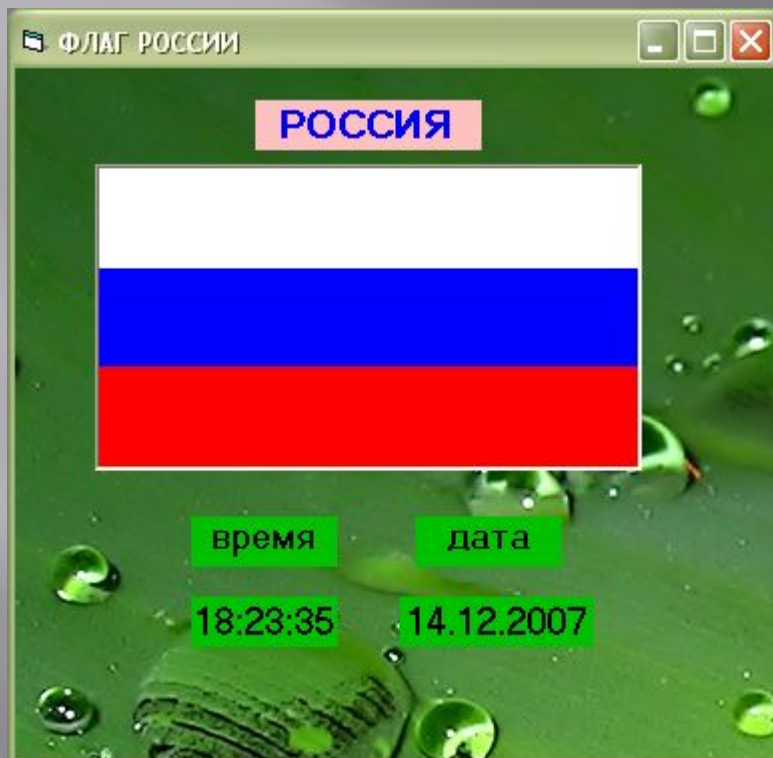
Установка системного времени

Установка текущей даты

назад

впер  
ед

Теперь запустим программу на выполнение и посмотрим как она работает:



Значения времени и даты появляются автоматически при запуске проекта.

← назад

→ вперед



# Сохранение проекта

1) Сохранение проекта. Так как проекты включают в себя несколько файлов (в том числе несколько файлов форм), то рекомендуется для каждого проекта создать отдельную папку на диске. Сохранение проекта производится с помощью меню *File*. Сначала необходимо сохранить форму и связанный с ней программный модуль с помощью пункта меню *Save Form1 As...* По умолчанию для файла формы предлагается имя, заданное в качестве значения свойства Name и расширение frm. Далее, необходимо сохранить файл проекта с помощью пункта меню *Save Project As...* В соответствии с соглашением об именах объектов целесообразно сохранить проект под именем с префиксом prj, например prjVB1.vbp.

2) Компиляция проекта в приложение. Сохраненный проект может выполняться только в самой системе программирования Visual Basic. Для того чтобы преобразовать проект в приложение, которое может выполняться непосредственно в среде операционной системы, необходимо сохранить проект в исполняемом файле (типа exe). Для компиляции проекта в исполняемый файл используется команда *Make Project1.exe...*

← назад

вперед →

На этом базовый уровень ознакомления с программой Visual Basic можно завершить.

Теперь вы сами можете разрабатывать некоторые программы.



назад



конец

Ц