

**ОСНОВЫ
ОБЪЕКТНО-
ОРИЕНТИРОВАННОГО
ВИЗУАЛЬНОГО
ПРОГРАММИРОВАНИЯ**

Проверочная работа

I вариант

1. Выберите целочисленные переменные:

a) single; б) string; в) short; г) integer; д) double

2. Выберите строковую переменную:

a) byte; б) string; в) single; г) boolean; д) integer

3. Напишите форматы тригонометрические функции.

4. Выберите функцию вырезания произвольной подстроки:

a) left; б) right; в) mid; г) len

5. Напишите формат функции ввода через диалоговое окно.

II вариант

1. Выберите переменные с плавающей запятой:

2. Выберите логическую переменную:

3. Напишите математические функции: квадратный корень и случайное число

4. Выберите функцию определения длины строки:

5. Напишите формат функции вывода сообщений с помощью окна сообщений.

Проект (*Project*)



Графический интерфейс

Основа – объект *форма* (окно), на нем размещаются другие объекты – *элементы управления*

Элементы управления:

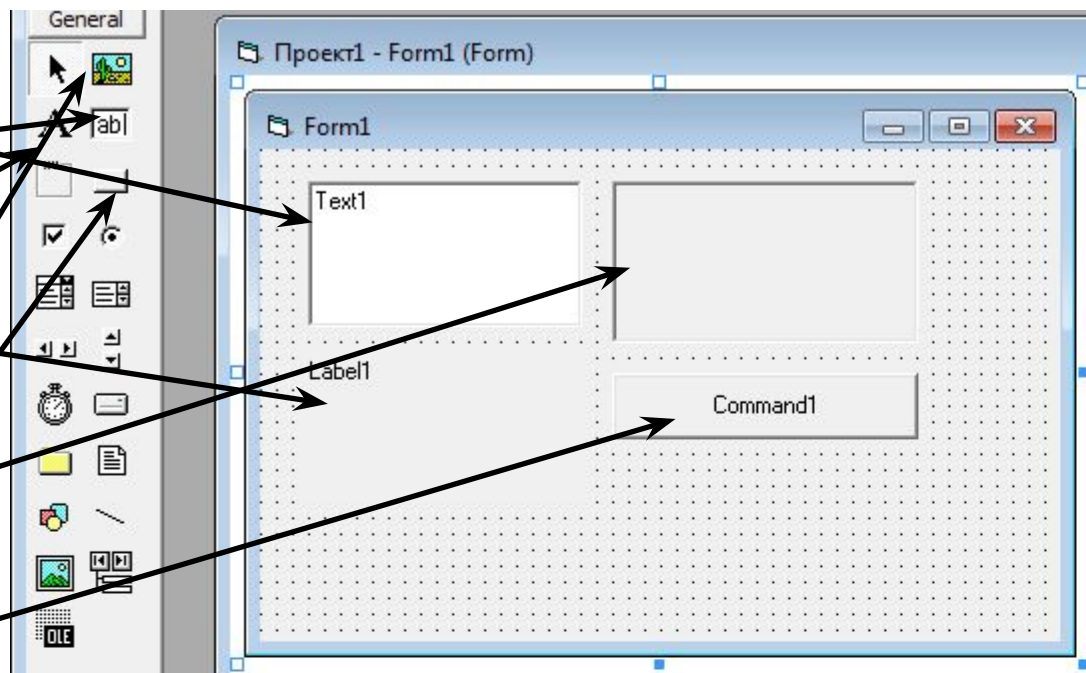
Текстовое поле (TextBox)

– для ВВОДА и ВЫВОДА
ДАННЫХ

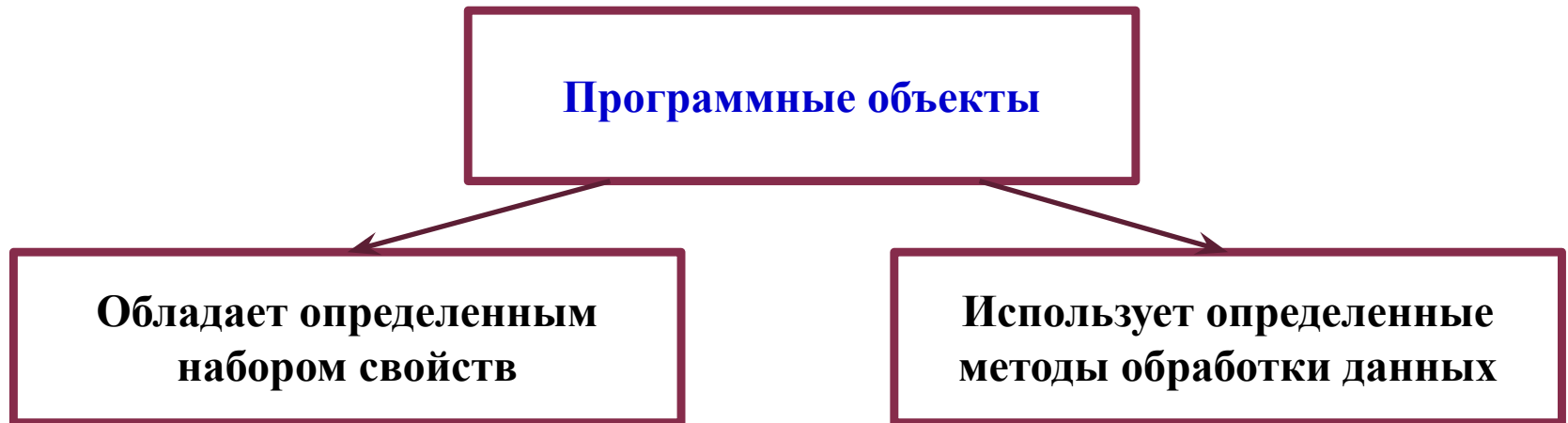
Метка (Label) – для
ВЫВОДА ДАННЫХ И
ПОЯСНИТЕЛЬНЫХ ТЕКСТОВ

**Графическое окно
(PictureBox)** – для ВЫВОДА
ГРАФИКИ

Кнопки (Button) – для
ЗАПУСКА ОБРАБОТЧИКОВ
СОБЫТИЙ



Объекты (*Objects*)



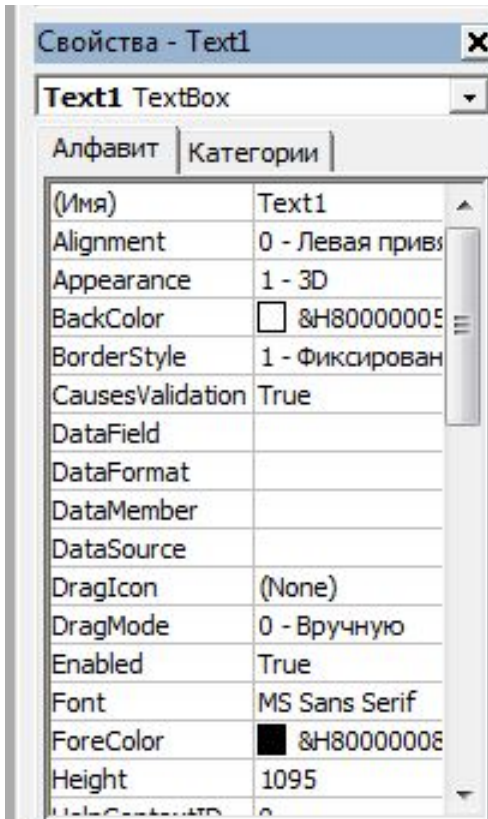
Объекты могут реагировать на *внешние события*.

Классы объектов – «шаблоны», определяющие наборы свойств, методов и событий для создания объектов.

Такой объект является *экземпляром класса* и наследует весь набор свойств, методов и событий данного класса

Например: класс объектов *форма (Form)* – основа для создания графического интерфейса служит шаблоном для экземпляров объектов форма: *Form1, Form2* и т.д.

Свойства объекта (*Properties*)



Каждый класс объектов обладает определенным набором свойств. Первоначальные значения свойств объектов можно установить с использованием диалогового окна Свойства (Properties). Значение свойств можно изменить в программном коде:

Объект.Свойство = ЗначениеСвойства

Например:

Form1.Text = "Первый проект"

Методы объекта (*Methods*)

Объекты могут использовать различные методы обработки данных. Методы имеют аргументы, которые позволяют задать параметры выполняемых действий.

Для использования метода в строке программного кода указывают имя объекта, через точку – метод, в скобках - аргументы:

Объект.Метод (арг1, арг2)

Например:

Form.Scale (2, 0.5)

Метод Scale (x, y) меняет размеры формы

Событие (*Event*)

Событие представляет собой действие, распознаваемое элементом управления. Событие может создаваться пользователем (щелчок мыши или нажатие клавиши) или быть результатом воздействия других программных объектов. Каждый объект реагирует на определенный набор событий.

Например:

Кнопка реагирует на:

- Щелчок кнопки мыши (*Click*)
- Нажатие кнопки мыши (*MouseDown*)
- Отпускание кнопки мыши (*MouseUp*)
- Нажатие клавиши (*KeyPress*)

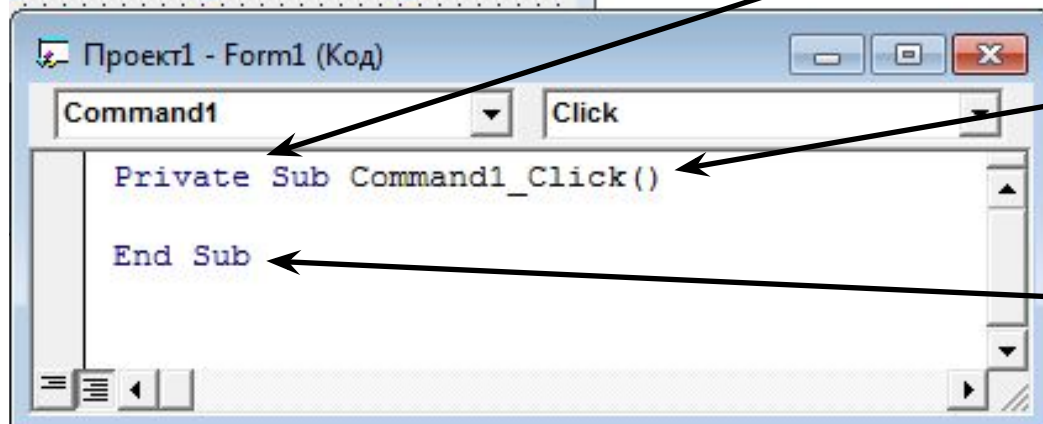
Обработчик события

Для каждого события можно запрограммировать *отклик* – реакцию объекта на произошедшее событие. В качестве отклика выполняется *обработчик события* (событийная процедура) – программа.

Чтобы создать заготовку обработчика события, необходимо осуществить *двойной щелчок* мышью по объекту:



Начало обработчика событий

A screenshot of the Visual Studio code editor showing the event handler for the "Command1" button. The code is as follows:

```
Private Sub Command1_Click()  
  
End Sub
```

The code is displayed in a window titled "Проект1 - Form1 (Код)". The "Command1" button and "Click" event are selected in the top-left corner of the editor.

Имя объекта и имя события

Конец обработчика событий

Обработчик события представляет собой программу, которая начинает выполняться после реализации определенного события