

Лабораторная работа 2 (4 часа)

Тема: Флажки, переключатели, контейнеры. Ветвления.

Цель:

- Изучить компоненты CheckBox, RadioButton, RadioGroup, Panel, GroupBox ;
- Закрепить навыки работы с условными операторами в Си Шарп

Содержание отчета (по каждому заданию)

1. Постановка задачи, пояснения при необходимости (формаализация)
2. Макеты форм в режиме конструирования с именами компонентов и их назначением
3. Таблица значений свойств компонентов, устанавливаемых в режиме конструирования
4. Программный код (процедуры - обработчики событий)
5. Протокол работы приложения (форма в режиме выполнения)

Замечание: Не забывайте в заголовке формы указывать ФИО, номер работы и задания!!!!!!!!!!!!

Краткие теоретические сведения:

Для выбора одного или нескольких вариантов действий (объектов) из предложенных существует ряд специальных элементов управления, к простейшим из которых относятся флажки и переключатели (радиокнопки). Рассмотрим их.

Компонент CheckBox

Компонент **CheckBox** - это флажок (независимый переключатель). Флажок CheckBox используется в программе для визуализации состояний включено-выключено. Каждый щелчок мышкой по компоненту CheckBox меняет его состояние на противоположное. Смена состояния переключателя также доступна и в программе.

CheckBox представляет собой сочетание небольшого окошка, которое и визуализирует наличием или отсутствием "галочки" состояние компонента (рис. 1). Основное свойство компонента CheckBox – **Checked** типа Boolean. То есть, на приведённом рисунке

```
CheckBox1.Checked = False;
```

CheckBox2.Checked = True.

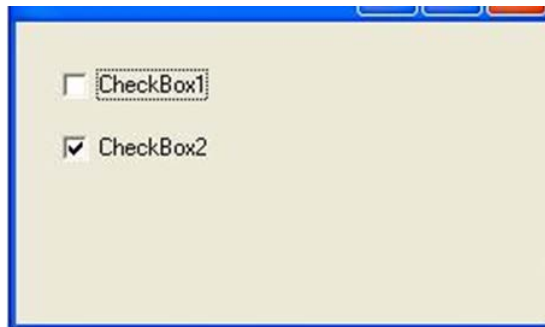


Рис. 1 Иллюстрация состояния флажка

Компонент **CheckBox** - независимый переключатель, то есть в группе из нескольких компонентов каждый из них может быть установлен в произвольное состояние, независимое от состояния остальных компонентов группы.

В программном коде с помощью условного оператора обычно анализируется, в каком состоянии находится флажок (выбор пользователя) и выполняются различные действия в зависимости от выбора:

If CheckBox1.Checked

{ <действия при установленном флажке> }

Else

{ <действия при сброшенном флажке> . }

Компонент RadioButton

Компонент **RadioButton** (рис. 2) - это "радиокнопка", и как следует из названия, служит для "переключения каналов", как и в настоящем радиоприёмнике. Компонент состоит из круглого окошка и текстовой строки. В окошке появляется отметка выбора данного компонента, текстовая строка поясняет его смысл. За изменение значения текстовой строки отвечает свойство **Text**.



Рис. 2 Иллюстрация компонента RadioButton

В отличие от **CheckBox**, компоненты **RadioButton** представляют собой **зависимые переключатели**, предназначенные для **выбора одного из нескольких взаимоисключающих решений**. На форму (точнее, в компонент-контейнер) помещается по меньшей мере два таких компонента. Они могут находиться в одном из двух состояний, определяемом свойством **Checked** (т.е. "выбрано") типа **Boolean**. У выбранного компонента свойство **Checked** равно **True**. Если в одном компоненте это свойство принимает значение **True**, то во всех других компонентах, расположенных в том же контейнере, свойства **Checked** принимают значение **False**.

Свойство **Checked** доступно как для чтения, так и для записи, то есть можно изменять значение этого свойства программно.

Элементы-контейнеры

В проектах часто возникает необходимость объединить несколько однородных управляющих элементов в группу. Для этой цели служат специальные элементы-контейнеры. Они могут выполнять как чисто декоративные функции, зрительно объединяя компоненты, связанные друг с другом по назначению, так и функции управления, организуя совместную работу своих дочерних компонентов. Рассмотрим некоторые из элементов-контейнеров.

Компонент **GroupBox**

Этот компонент служит контейнером для размещения дочерних компонентов и представляет собой прямоугольное окно с рамкой и текстом в разрыве рамки. Обычно с его помощью выделяется группа управляющих элементов, объединенных по функциональному назначению.

Компонент **Panel**

Компонент Panel (панель) представляет собой контейнер общего назначения. В отличие от GroupBox, он не имеет заголовка и поэтому менее удобен для функционального группирования компонентов. Компоненты этого класса часто помещают на форму для того, чтобы располагать вставленные в них дочерние компоненты вдоль одной из сторон окна независимо от изменения размеров этого окна.

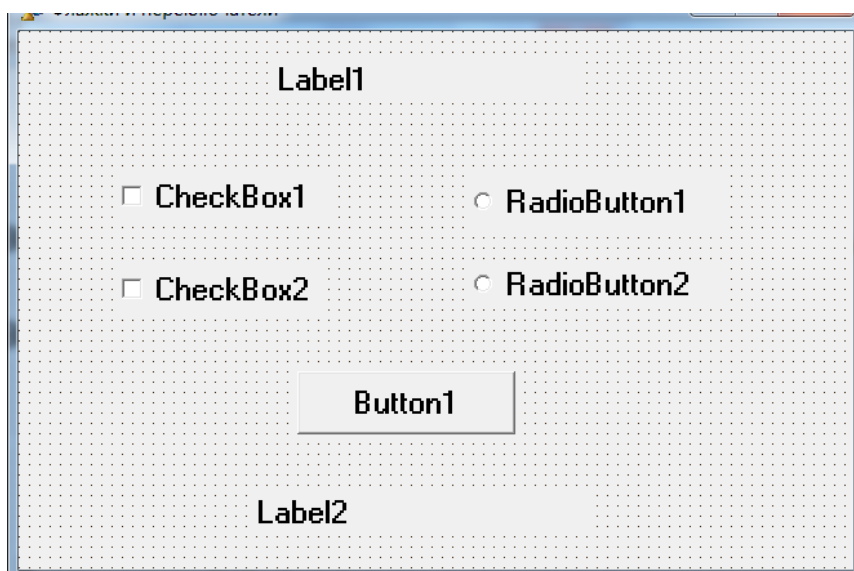
Задание 1. Иллюстрация работы с флажками и переключателями (для всех вариантов одинаковое)

Постановка задачи:

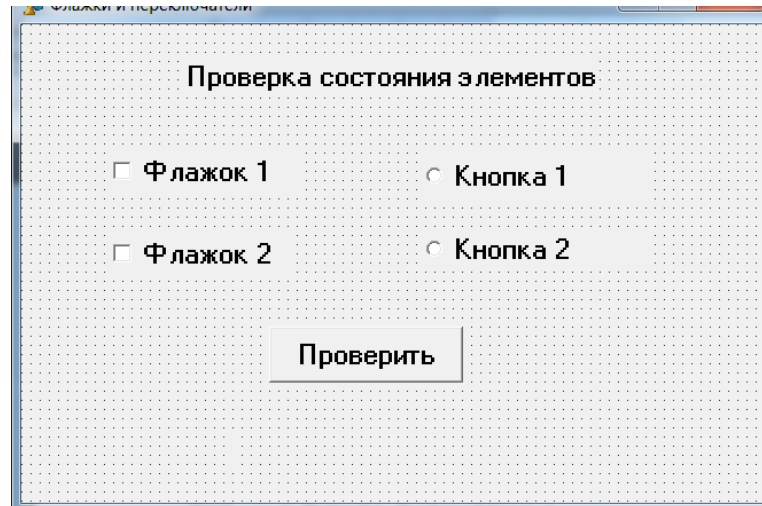
Создадим проект, состоящий из двух зависимых (RadioButton) и двух независимых (CheckBox) переключателей. С помощью командной кнопки «Проверить» будем определять, какие из них выбраны пользователем. При каждой проверке необходимо, чтобы выдавалось соответствующее сообщение о состоянии флажков и переключателей. Сообщение для независимых переключателей будем выводить в компонент Label. Для зависимых переключателей сообщение будем выводить непосредственно рядом с ними, в их надписях.

Пояснения-иллюстрации к заданию:

Форма в режиме конструирования (до задания свойств элементам)



Форма в режиме конструирования (после задания свойств элементам)



Для независимых переключателей возможны 4 варианта:

1. Оба флажка выбраны
2. Выбран первый флажок
3. Выбран второй флажок
4. Ни один из флажков не выбран

Создать обработчик события Click для компонента Button1 (выполним двойной щелчок по кнопке «проверить») и запишем в него соответствующий фрагмент программного кода для проверки этих четырех вариантов:

Теперь перейдем к зависимым переключателям. Их состояние будем отображать сразу при щелчке пользователя по элементу, используя для этого обработчик событий Click соответствующей радиокнопки. То есть создадим обработчик события Click для RadioButton1 и обработчик события Click для компонента RadioButton2.

Протокол работы программы для трех случаев представлен на рисунках ниже. Вам надо в отчете представить все 4 случая (добавить еще один – выбран 2 флажок) :

Проверка состояния элементов

Флажок 1 Выбран

Флажок 2 Не выбран

Выбраны оба флажка

Протокол работы (вариант 1)

Проверка состояния элементов

Флажок 1 Не выбран

Флажок 2 Выбран

Ни один флажок не выбран

Протокол работы (вариант 2)

Проверка состояния элементов

Флажок 1 Не выбран

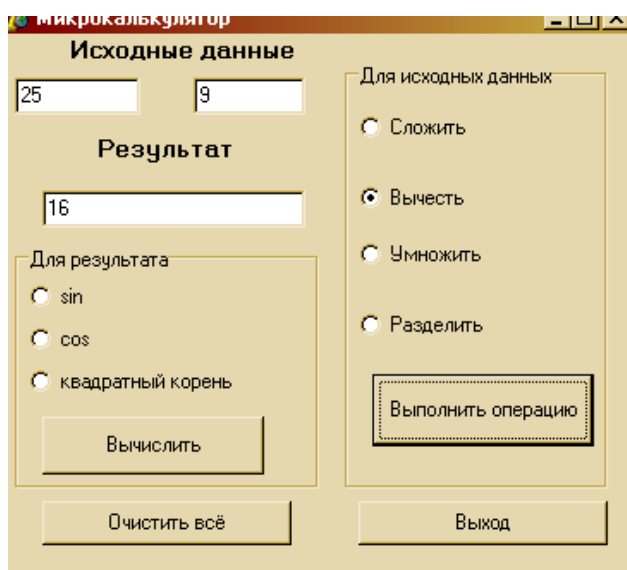
Флажок 2 Выбран

Выбран флажок 1

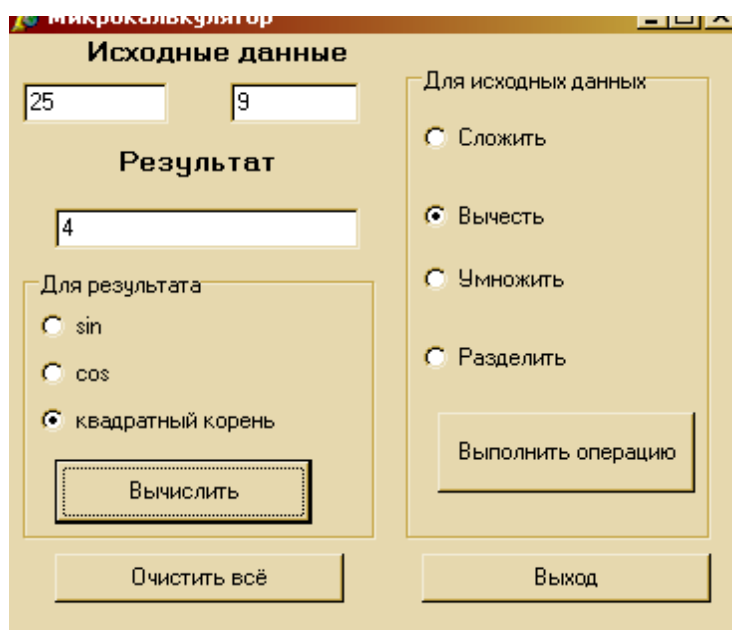
Протокол работы (вариант 3)

Задание 2 (для всех вариантов)

Постановка задачи: Создать приложение «Микрокалькулятор», позволяющее выполнять арифметические операции над двумя заданными числами и вычислять значения некоторых функций от полученного результата, например $\sin()$, $\cos()$, квадратный корень (желательно придумать свои функции). Для выбора операции использовать радиокнопки, для ввода исходных данных и вывода результата – текстовые поля (см. рисунок – протокол работы). Для вычислений использовать обработчики события Click кнопок «Выполнить операцию» и «Вычислить» соответственно:



Приложение «Микрокалькулятор» после нажатия на кнопку «Выполнить операцию».



Приложение «Микрокалькулятор» после нажатия на кнопку «Вычислить».

Задание 3.

Создать приложение «Тест по информатике», содержащий не менее трёх вопросов. Вопросы должны быть трех типов. Первый и второй - закрытого типа: 1) с выбором одного правильного ответа из предложенных, 2) с выбором нескольких правильных ответов. Третий – открытого типа с вводом ответа. Для организации вопросов первого типа используйте компоненты RadioButton, для второго - CheckBox, для третьего типа – TextBox. Вывод результатов-комментариев по каждому вопросу - Label. Для группировки вопросов используйте элемент GroupBox. Примерный вид формы в режиме конструирования приведен на рисунке :

Тест по информатике

Вопрос №1
Для вывода данных в Делфи используются элементы

Button Label
 Panel Edit

Проверить

Вопрос №2
Действие, распознаваемое объектом

отклик метод
 свойство событие.

Проверить

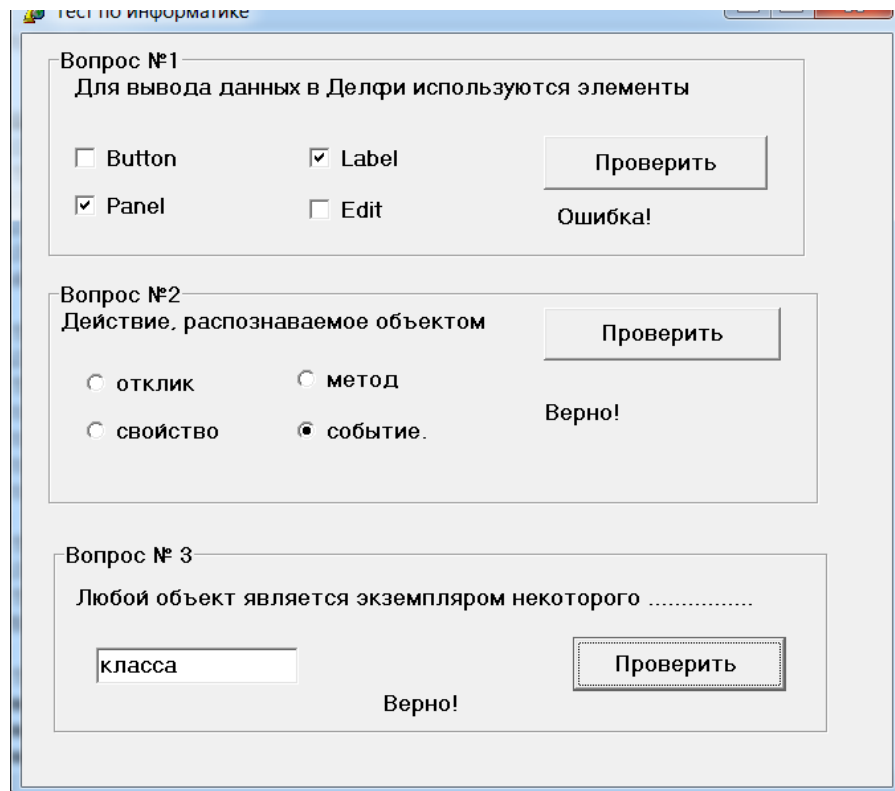
Вопрос №3
Любой объект является экземпляром некоторого

Проверить

Форма в режиме конструирования

При проверке правильности вопроса с несколькими вариантами ответов необходимо учесть, что надо обязательно проверить состояние каждого флажка, а не только правильных!!!!!!

На следующем рисунке приведен протокол работы приложения:



Приложение в режиме выполнения