

ЛАБОРАТОРНАЯ РАБОТА № 1.

ИЗУЧЕНИЕ СРЕДЫ РАЗРАБОТКИ VISUAL STUDIO

Цель лабораторной работы: изучить среду быстрой разработки приложений Visual Studio. Научиться размещать и настраивать внешний вид элементов управления на форме.

1.1. Интегрированная среда разработчика Visual Studio

Среда Visual Studio визуально реализуется в виде одного окна с несколькими панелями инструментов. Количество, расположение, размер и вид панелей может меняться программистом или самой средой разработки в зависимости от текущего режима работы среды или пожеланий программиста, что значительно повышает производительность работы.

При запуске Visual Studio появляется начальная страница со списком последних проектов, а также командами *Создать _____* и *Открыть _____*. Нажмите ссылку *Создать _____* или выберите в меню *Файл* команду *Создать _____*, на экране появится диалог для создания нового проекта (рис. 1.1).

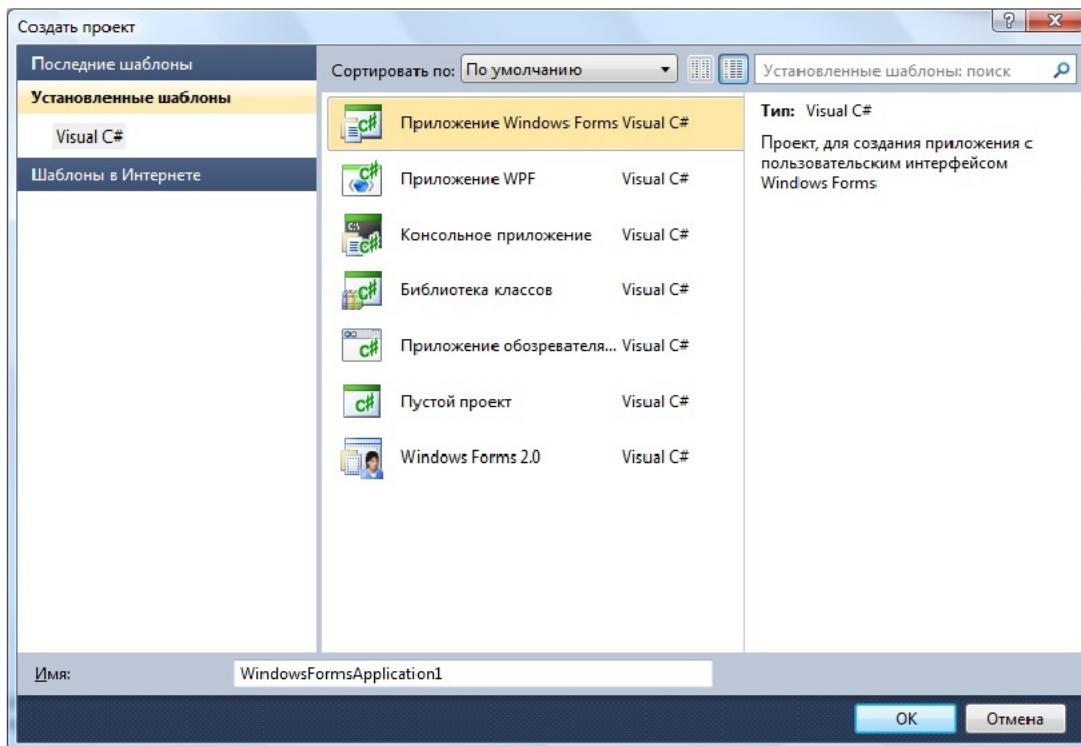


Рис. 1.1. Диалог создания нового проекта

Слева в списке шаблонов приведены языки программирования, которые поддерживает данная версия Visual Studio: убедитесь, что там выделен раздел Visual C#. В средней части приведены типы проектов, которые можно создать. В наших лабораторных работах будут использоваться два типа проектов:

1. *Приложение Windows Forms* –
-
-

2. *Консольное приложение* –
-
-

Выберите в списке тип проекта «Приложение Windows Forms», в поле «имя» внизу окна введите желаемое имя проекта (например, MyFirstApp) и нажмите кнопку ОК. Через несколько секунд Visual Studio создаст проект и Вы сможете увидеть на экране картинку, подобную представленной на рис. 1.2.

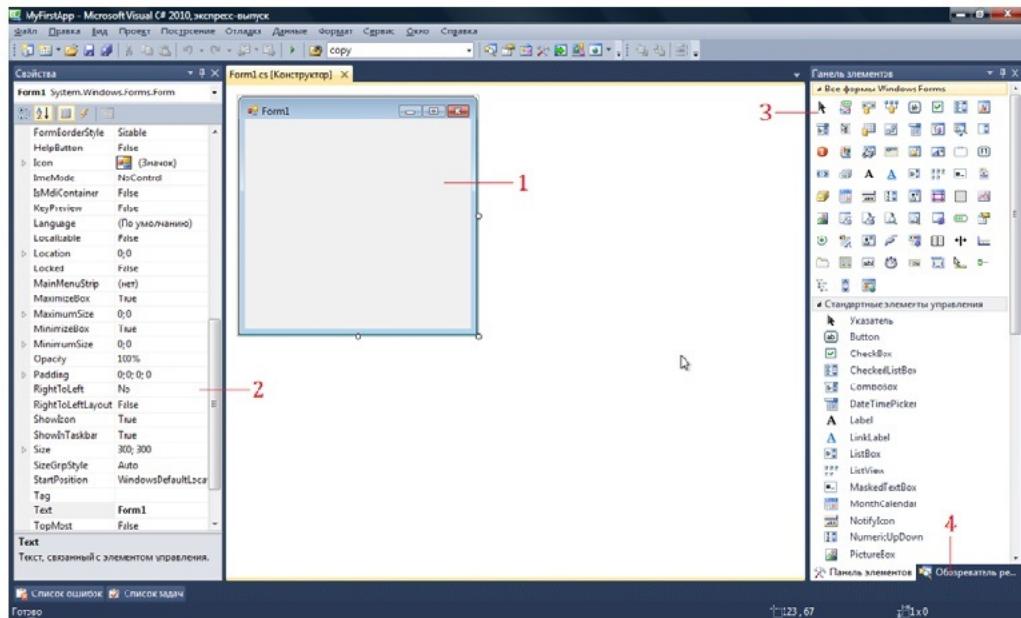


Рис. 1.2. Главное окно Visual Studio

В главном окне Visual Studio присутствует несколько основных элементов, которые будут помогать нам в работе. Прежде всего, это **форма** (1) – будущее окно нашего приложения, на котором будут размещаться элементы управления. При выполнении программы помещенные элементы управления будут иметь тот же вид, что и на этапе проектирования.

Второй по важности объект – это **окно свойств** (2), в котором _____

С помощью кнопки можно просматривать свойства элемента управления, а кнопка переключает окно в режим просмотра событий. Чтобы было удобнее искать нужные свойства, можно отсортировать их по алфавиту с помощью кнопки . Если этого окна на экране нет, его можно активировать в меню *Вид* → *Окно свойств* (иногда этот пункт вложен в подпункт *Другие окна*).

Сами элементы управления можно брать на **панели элементов** (3). Все элементы управления разбиты на логические группы, что облегчает поиск нужных элементов. Если панели нет на экране, ее нужно активировать командой *Вид* → *Панель элементов*.

Наконец, **обозреватель решений** (4) содержит список всех файлов, входящих в проект, включая добавленные изображения и служебные файлы. Активируется командой *Вид* → *Обозреватель решений*.

Указанные панели могут уже присутствовать на экране, но быть скрытыми за другими панелями или свернуты к боковой стороне окна. В этом случае достаточно щелкнуть на соответствующем ярлычке, чтобы вывести панель на передний план.

Окно текста программы предназначено для просмотра, написания и редактирования текста программы. Переключаться между формой и текстом программы можно с помощью команд *Вид* → *Код* и *Вид* → *Конструктор*. При первоначальной загрузке в окне текста программы находится текст, содержащий минимальный набор операторов для нормального функционирования пустой формы в качестве Windows-окна. При помещении элемента управления в окно формы, текст программы автоматически дополняется описанием необходимых для его работы библиотек стандартных программ (раздел *using*) и переменных для доступа к элементу управления (в скрытой части класса формы).

Программа на языке C# составляется как описание алгоритмов, которые необходимо выполнить, если возникает определенное событие, связанное с формой (например, щелчок «мыши» на кнопке – событие *Click*, загрузка формы – *Load*). Для каждого обрабатываемого в форме события, с помощью окна свойств, в тексте программы организуется метод, в котором программист записывает на языке C# требуемый алгоритм.

1.2. Настройка формы

Настройка формы начинается с настройки размера формы. С помощью мыши, «захватывая» одну из кромок формы или выделенную строку заголовка, отрегулируйте нужные размеры формы.

Для настройки будущего окна приложения задаются свойства формы. Для задания любых свойств формы и элементов управления на форме используется окно свойств.

Новая форма имеет одинаковые имя (*Name*) и заголовок (*Text*) – *Form1*.

Для изменения заголовка щелкните кнопкой мыши на форме, в окне свойств найдите и щелкните мышкой на строчке с названием *Text*. В выделенном окне наберите «*Лаб. раб. N1. Ст. гр. 7А62 Иванов А. А.*». Для задания цвета окна используйте свойство *BackColor*.

1.3. Размещение элементов управления на форме

Для размещения различных элементов управления на форме используется панель элементов. Панель элементов содержит элементы управления, сгруппированные по типу. Каждую группу элементов управления можно свернуть, если она в настоящий момент не нужна. Для выполнения лабораторных работ потребуются элементы управления из группы *Стандартные элементы управления*.

Щелкните на элементе управления, который хотите добавить, а затем щелкните в нужном месте формы – элемент появится на форме. Элемент можно перемещать по форме, схватившись за него левой кнопкой мышки (иногда это можно сделать лишь за появляющийся при нажатии на элемент квадрат со стрелками). Если элемент управления позволяет изменять размеры, то на соответствующих его сторонах появятся белые кружки, ухватившись за которые и можно изменить размер. После размещения элемента управления на форме, его можно выделить щелчком мыши и при этом получить доступ к его свойствам в окне свойств.

1.4. Размещение строки ввода

Если необходимо ввести из формы в программу или вывести на форму информацию, которая вмещается в одну строку, используют окно одностороннего редактора текста, представляемого элементом управления *TextBox*.

В данной программе с помощью одностороннего редактора будет вводиться имя пользователя.

Выберите на панели элементов пиктограмму с названием *TextBox*, щелкните мышью в том месте формы, где вы хотите ее поставить. Захватив его мышкой, отрегулируйте размеры элемента управления и его положение. Обратите внимание на то, что теперь в тексте программы можно использовать переменную *textBox1*, которая соответствует добавленному элементу управления. В этой переменной в свойстве *Text* будет содержаться строка символов (тип *string*) и отображаться в соответствующем окне *TextBox*.

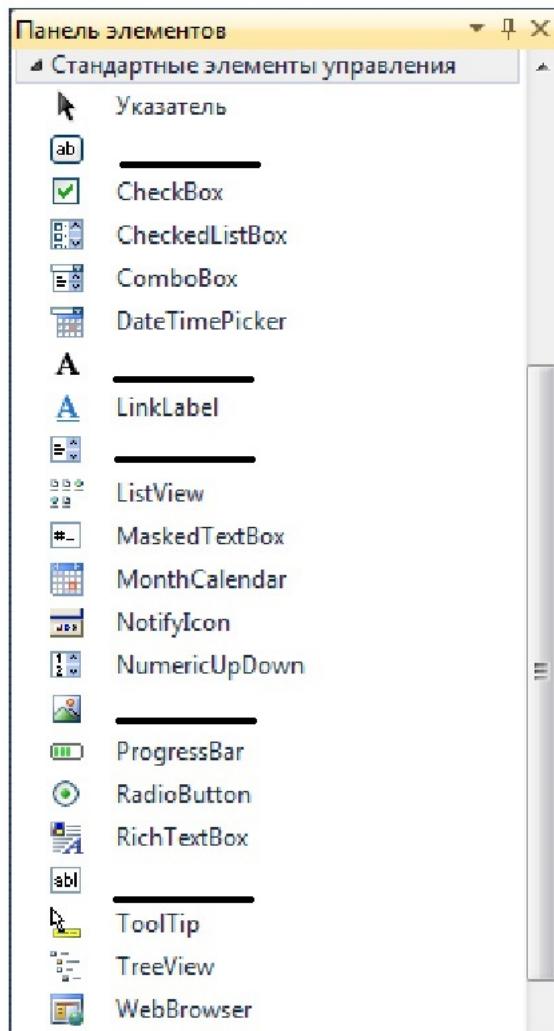


Рис. 1.3. Панель элементов

С помощью окна свойств установите шрифт и размер символов, отражаемых в строке TextBox (свойство Font).

1.5. Размещение надписей

На форме могут размещаться пояснительные надписи. Для нанесения таких надписей на форму используется элемент управления Label. Выберите на панели элементов пиктограмму с названием Label, щелкните на ней мышью. После этого в нужном месте формы щелкните мышью, появится надпись label1. Щелкнув на ней мышью, отрегулируйте размер и, изменив свойство Text в окне свойств, введите строку, например «*Введите свое имя:*», а также выберите размер символов (свойство Font).

Обратите внимание, что в тексте программы теперь можно обращаться к новой переменной типа Label. В ней хранится пояснительная строка, которую можно изменять в процессе работы программы.

1.6. Написание программы обработки событий

С каждым элементом управления на форме и с самой формой могут происходить события во время работы программы. Например, с кнопкой может произойти событие – нажатие кнопки, а с окном, которое проектируется с помощью формы, может произойти ряд событий: создание окна, изменение размера окна, щелчок мыши на окне и т. п. Эти события могут обрабатываться в программе. Для обработки таких событий необходимо создать обработчики события – специальный *метод*. Для создания обработчика события существует два способа.

Первый способ – создать обработчик для события по умолчанию (обычно это самое часто используемое событие данного элемента управления). Например, для кнопки таким образом создается обработчик события нажатия.

1.7. Написание программы обработки события нажатия кнопки

Поместите на форму кнопку, которая описывается элементом управления Button. С помощью окна свойств измените заголовок (Text) на слово «Привет» или другое по вашему желанию. Отрегулируйте положение и размер кнопки.

После этого два раза щелкните мышью на кнопке, появится текст программы:

```
private void button1_Click(object sender, EventArgs e)
{
}
```

Это и есть обработчики события нажатия кнопки. Вы можете добавлять свой код между скобками { }. Например, наберите:

```
MessageBox.Show("Привет, " + textBox1.Text + "!");
```

1.8. Написание программы обработки события загрузки формы

Второй способ создания обработчика события заключается в выборе соответствующего события для выделенного элемента на форме. При этом используется окно свойств и его закладка . Рассмотрим этот способ. Выделите форму щелчком по ней, чтобы вокруг нее появилась рамка из точек. В окне свойств найдите событие Load. Щелкните по данной строчке дважды мышкой. Появится метод:

```
private void Form1_Load(object sender, EventArgs e)
{
```

}

Между скобками { } вставим текст программы:

```
BackColor = Color.AntiqueWhite;
```

Каждый элемент управления имеет свой набор обработчиков событий, однако некоторые из них присущи большинству элементов управления. Наиболее часто применяемые события описаны ниже:

- _____ форма получает это событие при активации.
- _____ возникает при загрузке формы. В обработчике данного события следует задавать действия, которые должны происходить в момент создания формы, например установка начальных значений.
- _____ возникает при нажатии кнопки на клавиатуре. Параметр `e.KeyChar` имеет тип `char` и содержит код нажатой клавиши (клавиша `Enter` клавиатуры имеет код #13, клавиша `Esc` – #27 и т. д.). Обычно это событие используется в том случае, когда необходима реакция на нажатие одной из клавиш.
- _____ возникает при нажатии клавиши на клавиатуре. Обработчик этого события получает информацию о нажатой клавише и состоянии клавиш `Shift`, `Alt` и `Ctrl`, а также о нажатой кнопке мыши. Информация о клавише передается параметром `e.KeyCode`, который представляет собой перечисление `Keys` с кодами всех клавиш, а информацию о клавишах-модификаторах `Shift` и др. можно узнать из параметра `e.Modifiers`.
- _____ является парным событием для `KeyDown` и возникает при отпускании ранее нажатой клавиши.
- _____ возникает при нажатии кнопки мыши в области элемента управления.
- _____ возникает при двойном нажатии кнопки мыши в области элемента управления.

Важное примечание! Если какой-то обработчик был добавлен по ошибке или больше не нужен, то для его удаления нельзя просто удалить программный код обработчика! Сначала нужно удалить строку с именем обработчика в окне свойств на закладке . В противном случае программа может перестать компилироваться и даже отображать форму в дизайнере Visual Studio.

1.9. Запуск и работа с программой

Запустить программу можно, выбрав в меню *Отладка* команду *Начать отладку*. При этом происходит трансляция и, если нет ошибок, компоновка программы и создание единого загружаемого файла с расширением .exe. На экране появляется активное окно программы.

Если в программе есть ошибки, то в окне Список ошибок появляются найденные ошибки, а программа обычно не запускается. Иногда, однако, может быть запущена предыдущая версия программы, в которой еще нет последних изменений: чтобы этого не происходило, нужно в настройках Visual Studio установить опцию показа окна ошибок и запретить запуск при наличии ошибок (рис. 1.4 и 1.5).

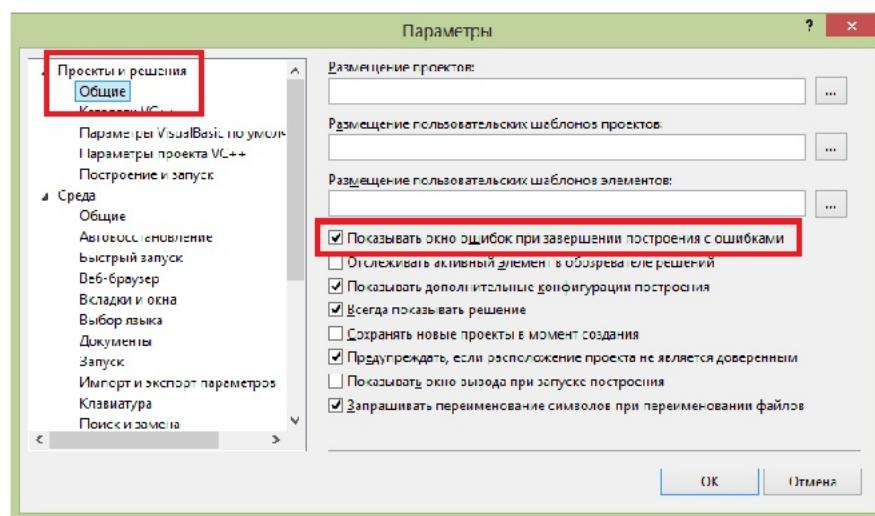


Рис. 1.4. Опция показа сообщений об ошибках

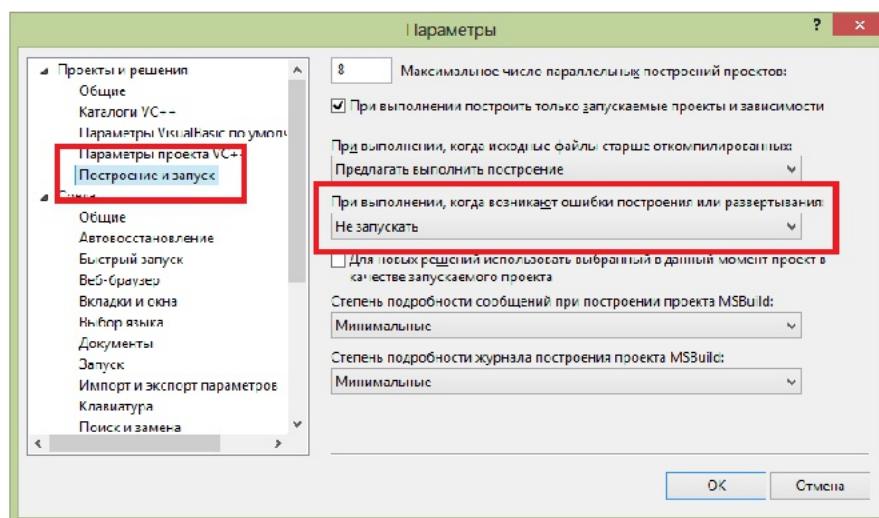


Рис. 1.5. Опция отключения запуска предыдущей версии при ошибках построения

Для завершения работы программы и возвращения в режим проектирования формы не забудьте закрыть окно программы!

1.10. Динамическое изменение свойств

Свойства элементов на окне могут быть изменены динамически во время выполнения программы. Например, можно изменить текст надписи или цвет формы. Изменение свойств происходит внутри обработчика события (например, обработчика события нажатия на кнопку). Для этого используют оператор присвоения вида:

```
<имя элемента>.<свойство> = <значение>;
```

Например:

```
label1.Text = "Привет";
```

<Имя элемента> определяется на этапе проектирования формы, при размещении элемента управления на форме. Например, при размещении на форме ряда элементов TextBox, эти элементы получают имена textBox1, textBox2, textBox3 и т. д., которые могут быть заменены в окне свойств в свойстве (Name) для текущего элемента. Допускается использование латинских или русских символов, знака подчеркивания и цифр (цифра не должна стоять в начале идентификатора). Список свойств для конкретного элемента можно посмотреть в окне свойств, а также в приложении к данным методическим указаниям.

Если требуется изменить свойства формы, то никакое имя элемента перед точкой вставлять не нужно, как и саму точку. Например, чтобы задать цвет формы, нужно просто написать:

```
BackColor = Color.Green;
```

1.11. Выполнение индивидуального задания

По указанию преподавателя выберите свое индивидуальное задание. Уточните условие задания, количество, наименование, типы исходных данных. Прочтите в Приложении 1 описание свойств и описание элементов управления Form, Label, TextBox, Button. С помощью окна свойств установите первоначальный цвет формы, шрифт выводимых символов.

Индивидуальные задания

1. Разместите на форме четыре кнопки (Button). Сделайте на кнопках следующие надписи: «красный», «зеленый», «синий», «желтый». Создайте четыре обработчика события нажатия на данные кноп-

ки, которые будут менять цвет формы в соответствии с текстом на кнопках.

2. Разместите на форме две кнопки (**Button**) и одну метку (**Label**). Сделайте на кнопках следующие надписи: «привет», «до свидания». Создайте обработчики события нажатия на данные кнопки, которые будут менять текст метки на слова, написанные на кнопках. Создайте обработчик события создания формы (**Load**), который будет устанавливать цвет формы и менять текст метки на строку «Начало работы».

3. Разместите на форме ряд кнопок (**Button**) напротив каждой поле ввода (**TextBox**) и одну метку (**Label**). Создайте обработчики события нажатия на данные кнопки, которые будут менять текст в метке. Текст в метке берется из поля ввода напротив нажимаемой кнопки.

4. Разместите на форме ряд кнопок (**Button**), и одно поле ввода (**TextBox**). Создайте обработчики события нажатия на данные кнопки, которые будут менять текст на нажатой кнопке. Текст на кнопке берется из поля ввода.

5. Разместите на форме ряд кнопок (**Button**) и ряд меток (**Label**). Создайте обработчики события нажатия на данные кнопки, которые будут менять цвет двух меток. Создайте обработчик события нажатия кнопки мыши на форме (**Click**), который будет устанавливать цвет всех меток в белый.

6. Разместите на форме ряд кнопок (**Button**) и ряд меток (**Label**). Создайте обработчик события создания формы (**Load**), который будет делать все метки невидимыми. Создайте обработчики события нажатия на кнопки, которые будут менять свойство метки **Visible**, тем самым делать их видимыми.

7. Разместите на форме ряд кнопок (**Button**), напротив каждой поле ввода (**TextBox**). Создайте обработчики события нажатия на данные кнопки, которые будут менять заголовок окна. Текст в заголовке берется из поля ввода напротив нажимаемой кнопки.

8. Разместите на форме две кнопки (**Button**) и одну метку (**Label**). Сделайте на кнопках следующие надписи: «скрыть», «показать». Создайте обработчики события нажатия на данные кнопки, которые будут скрывать или показывать метку. Создайте обработчик события создания формы (**Load**), который будет устанавливать цвет формы и менять текст метки на строку «Начало работы».

9. Разместите на форме три кнопки (**Button**) и одно поле ввода (**TextBox**). Сделайте на кнопках следующие надписи: «скрыть», «показать», «очистить». Создайте обработчики события нажатия на данные кнопки, которые будут скрывать или показывать поле ввода. При нажатии на кнопку «очистить» текст из поля ввода должен быть удален.

10. Разместите на форме две кнопки (**Button**) и одно поле ввода (**TextBox**). Сделайте на кнопках следующие надписи: «заполнить», «очистить». Создайте обработчики события нажатия на данные кнопки, которые будут очищать или заполнять поле ввода знаками «*****». Создайте обработчик события создания формы (**Load**), который будет устанавливать цвет формы и менять текст в поле ввода на строку «+++++».

11. Разработайте игру, которая заключается в следующем. На форме размещены пять кнопок (**Button**). При нажатии на кнопку некоторые кнопки становятся видимыми, а другие – невидимыми. Цель игры – скрыть все кнопки.

12. Разработайте игру, которая заключается в следующем. На форме размещены четыре кнопки (**Button**) и четыре метки (**Label**). При нажатии на кнопку часть надписей становится невидимой, а часть, наоборот, становится видимой. Цель игры – скрыть все надписи.

13. Разместите на форме ряд кнопок (**Button**). Создайте обработчики события нажатия на данные кнопки, которые будут делать неактивными текущую кнопку. Создайте обработчик события изменения размера формы (**Resize**), который будет устанавливать все кнопки в активный режим.

14. Разместите на форме ряд кнопок (**Button**). Создайте обработчики события нажатия на данные кнопки, которые будут делать неактивными следующую кнопку. Создайте обработчик события нажатия кнопки мыши на форме (**Click**), который будет устанавливать все кнопки в активный режим.

15. Разместите на форме три кнопки (**Button**) и одно поле ввода (**TextBox**). Сделайте на кнопках следующие надписи: «*****», «+++++», «00000». Создайте обработчики события нажатия на данные кнопки, которые будут выводить текст, написанный на кнопках, в поле ввода. Создайте обработчик события создания формы (**Load**), который будет устанавливать цвет формы и менять текст в поле ввода на строку «Готов к работе».

16. Разместите на форме ряд полей ввода (**TextBox**). Создайте обработчики события нажатия кнопкой мыши на данные поля ввода, которые будут выводить в текущее поле ввода его номер. Создайте обработчик события изменения размера формы (**Resize**), который будет очищать все поля ввода.

17. Разместите на форме поле ввода (**TextBox**), метку (**Label**) и кнопку (**Button**). Создайте обработчик события нажатия на кнопку, который будет копировать текст из поля ввода в метку. Создайте обработчик события нажатия кнопки мышки на форме (**Click**), который будет устанавливать цвет формы и менять текст метки на строку «Начало работы» и очищать поле ввода.

18. Разместите на форме поле ввода (**TextBox**) и две кнопки (**Button**) с надписями: «блокировать», «разблокировать». Создайте обработчики события нажатия на кнопки, которые будут делать активным или неактивным поле ввода. Создайте обработчик события нажатия кнопки мышки на форме (**Click**), который будет устанавливать цвет формы и делать невидимыми все элементы.

19. Реализуйте игру минер на поле 3×3 из кнопок (**Button**). Первоначально все кнопки не содержат надписей. При попытке нажатия на кнопку на ней либо показывается количество мин, либо надпись «мина!» и меняется цвет окна.

20. Разместите на форме четыре кнопки (**Button**). Напишите для каждой обработчик события, который будет менять размеры и местоположение на окне других кнопок.