C#

Windows Forms: КАЛЬКУЛЯТОР & МЕНЮ

Программная модель Windows Forms

- Программная модель Windows Forms определяет:
  - формы (окна и диалоги)
  - контролы (текстовые поля, кнопки, меню, ленты с инструментами, ...)
  - события, которыми управляют
  - жизненный цикл приложения
  - модель перерисовки контролов
  - управление фокусом и навигацией
- Жизненный цикл приложения основан на сообщениях
  - Контролы получают сообщения о действиях пользователя и реагируют по-своему

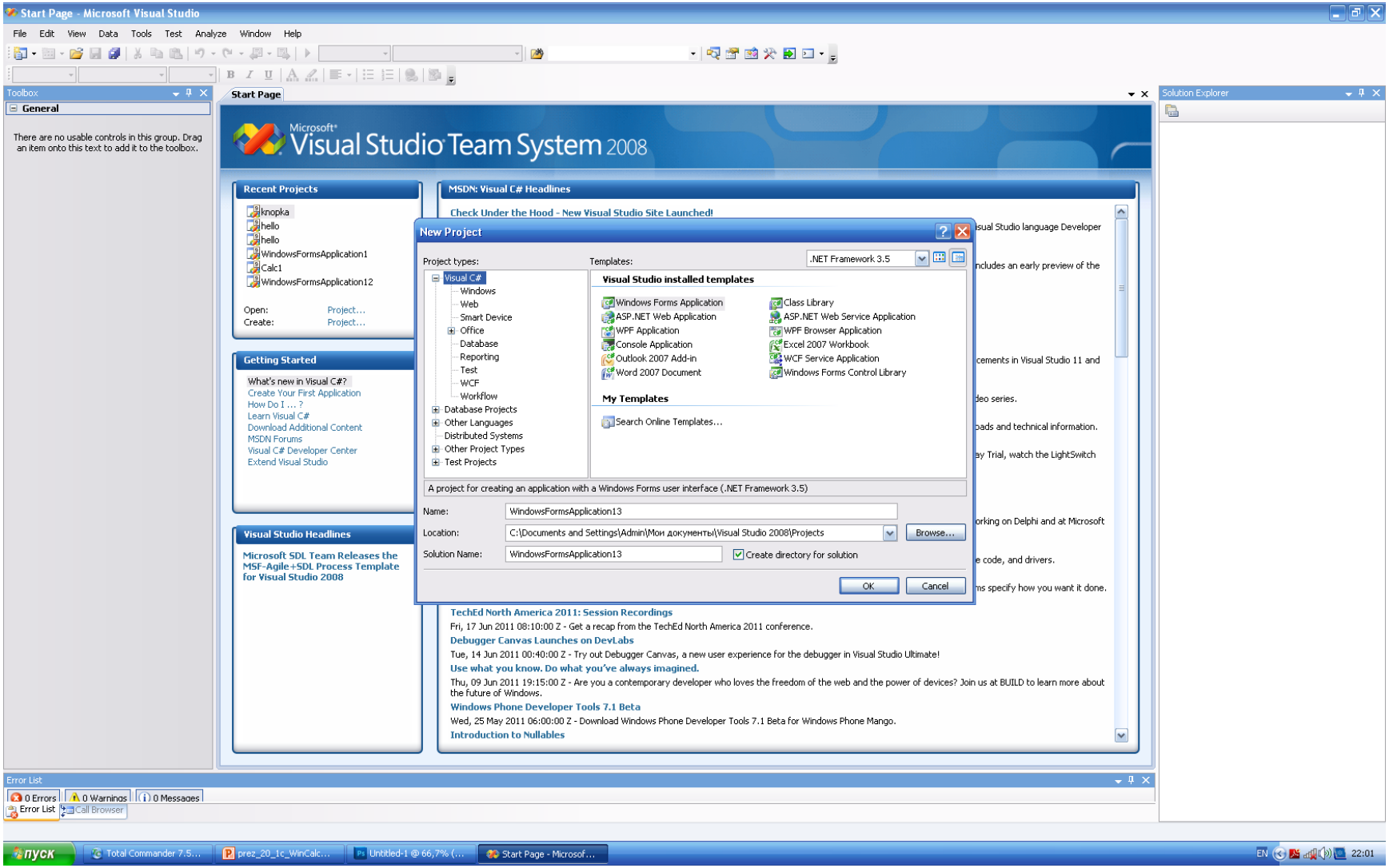
Парадигма работы Windows Forms приложения

- Постоянно слушает сообщения
- При получении сообщения (например, при перемещении мыши, при нажатии клавиш и т.д.) оно обрабатывается:
  - ищется контрол, к которому относится сообщение
  - ему передается сообщение
  - если контрол является контейнер-контролом, он ищет внутри себя, какому из содержащихся контролов предназначено сообщение и передает ему управление
- При закрытии главной формы приложения выполнение приложения прикращается

Этапы разработки приложений в Windows Forms

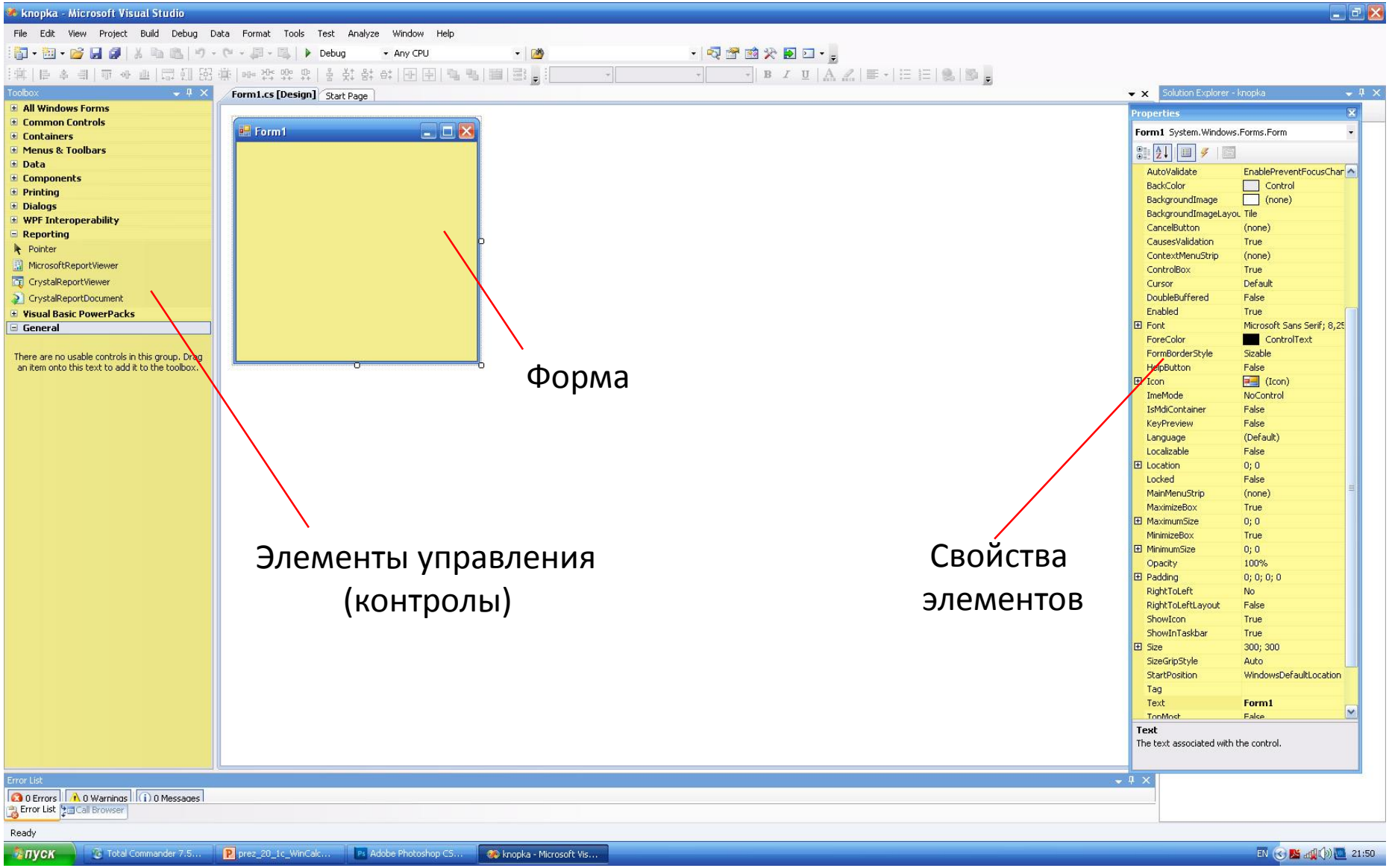
1. Уяснить и декомпозировать задачу
2. Построить структуры данных и алгоритмы
3. Разработать и утвердить эскиз интерфейса
4. Стартовать Visual Studio – Windows Forms
5. Видим: элементы управления – форма – свойства элементов
6. Выбираем нужные для решения элементы
7. Переносим элементы на форму
8. Устанавливаем свойства для каждого элемента
9. *Пишем обработчик событий для каждого элемента*

Старт Visual Studio для разработки Win-приложений

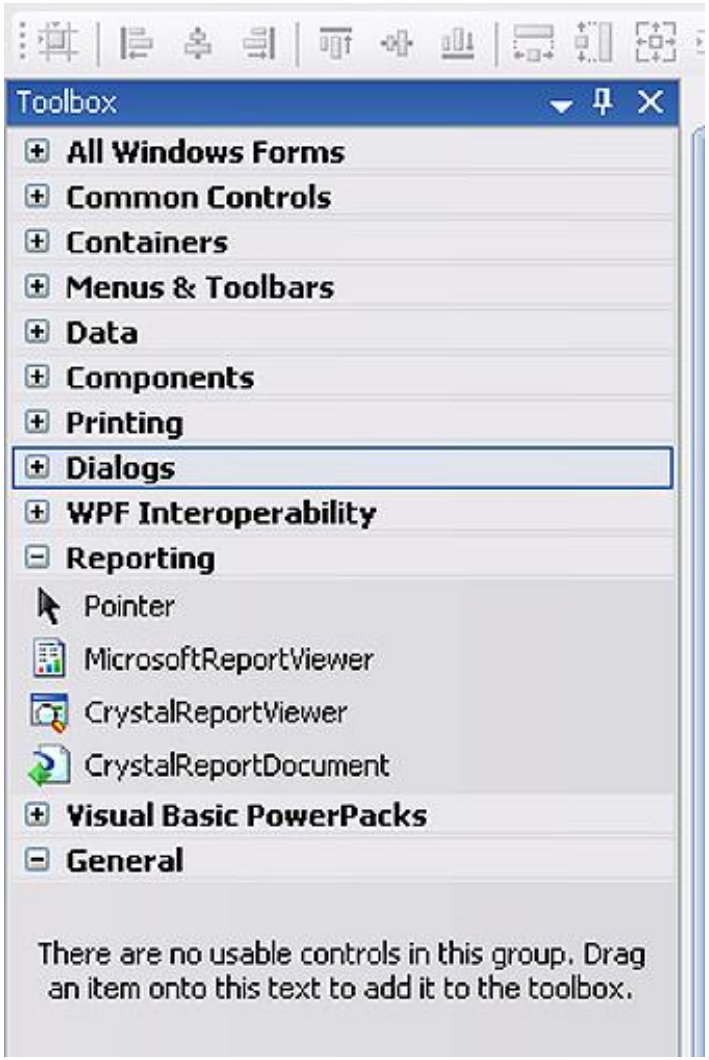


File – New – Project – Windows Forms Application – Name - экран разработки

Структура окна визуальной разработки



Палитра контролов (Tool Box)



All Windows forms- изначальный .Net набор, устарел, не рекомендуется
Common Controls – новые, современные контролы, рекомендуется
Containers – визуальная группировка группы контролов
Menus & Toolbars – меню, строки состояния и т.д.
Data - работа с базами данных
Components – набор полезных элементов (Timer, BackgroundWorker, HelpProvider)
Printing – управления печатью
Dialogs – типовые диалоги (OpenFileDialog,ColoDialog,)
WPF interoperability Reporting – доступ к ресурсам
Visual Basic Powerpack – доступ к ресурсам
General -

Основные контролы

- **TextBox** – 

– поле ввода текста.
  Важнейшие свойства:
  - Multiline – задает, можно ли вводить несколько строк
  - Text (Lines) – содержит введенный текст
- **Label** – 

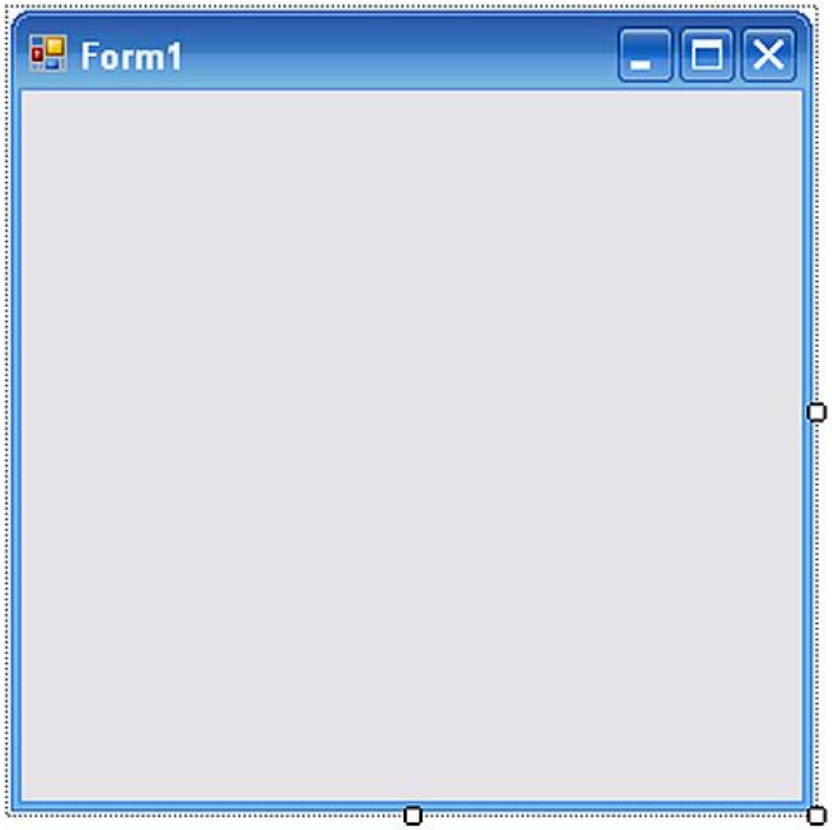
– отображает текст в форме.
  Важные свойства:
  - Text – текст, который отображается
- **Button** – 

– нажимающаяся кнопка.
  Важные свойства и события:
  - Click – активируется при нажатии
  - Text – задает текст на кнопке

Порядок работы с контролами

- Выбрать контрол в палитре
- Перетащить контрол на форму, контрол в фокусе
- Уточнить мышью местоположение и размер контрола на форме
- Перейти в палитру свойств
- Установить требуемые значения свойств контрола
- Добавить обработчик событий для контрола

Форма



Все формы наследуются от класса Form
Форма – тоже контрол, соответственно имеет свойства
Понятие фокуса
После старта форма автоматически в фокусе (рамка с квадратиками)
По умолчанию имя формы – Form1
Мы меняем его на имя нашего приложения. Разными способами
Лучше стандартизировать – через свойства Properties – Text

Палитра свойств контролов (Properties)

Важные свойства класса Control:

Name – имя контрола в программе

Text – надпись на кнопках

Controls – содержит коллекцию вложенных контролов (если имеет)

CanFocus – определяет, может ли контрол получать фокус

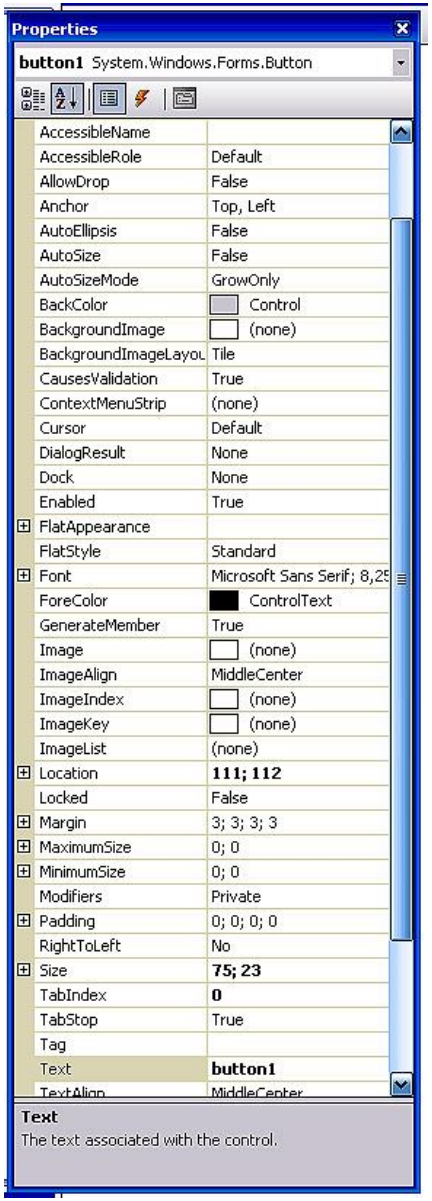
Enabled – позволяет "выключить" контрол (он остается видим, но не активен)

Font – задает шрифт (имя, стиль, размер)

ForeColor – задает цвет контрола

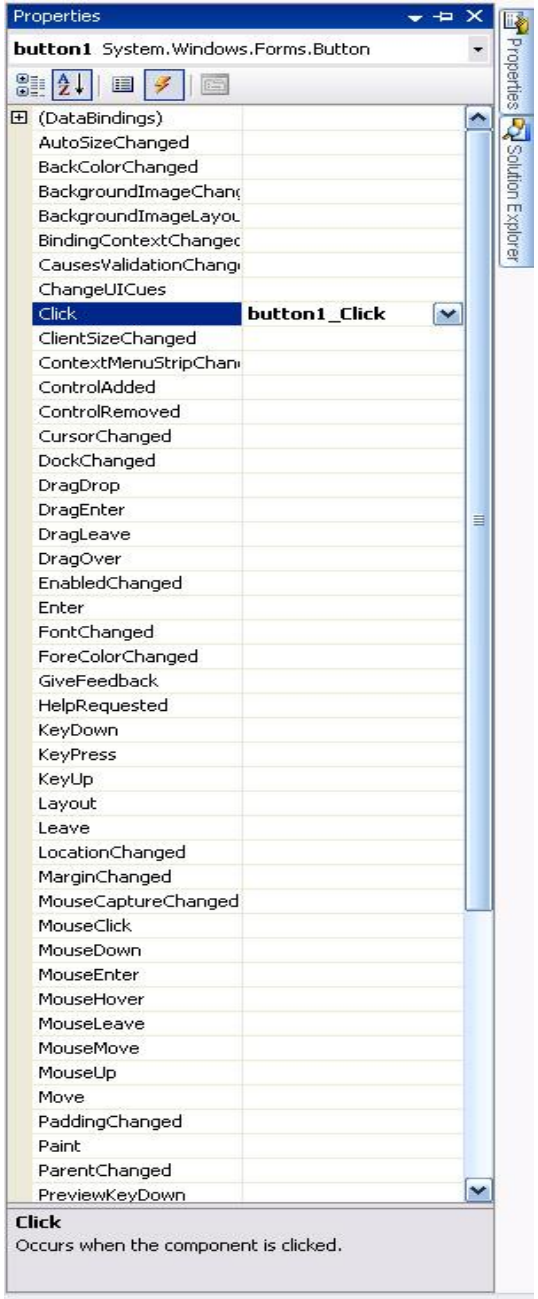
Location – содержит позицию контрола в своем контейнере

Parent – задает контейнер-контрол, содержащий текущий контрол



Важнейшие события

- Важные события класса Control:
  - **Click** – наступает при щелчке мышкой на контроле
  - **Enter, Leave** – наступает при активации и деактивации контрола
  - **KeyDown, KeyUp** – наступает при нажатии и отпускании клавиш (или их комбинации)
  - **KeyPress** – при нажатии на нефункциональную клавишу
  - **MouseDown, MouseUp, MouseHover, MouseEnter, MouseLeave, MouseMove, MouseWheel** – наступает при событиях от мышки, указатель которой находится поверх контрола

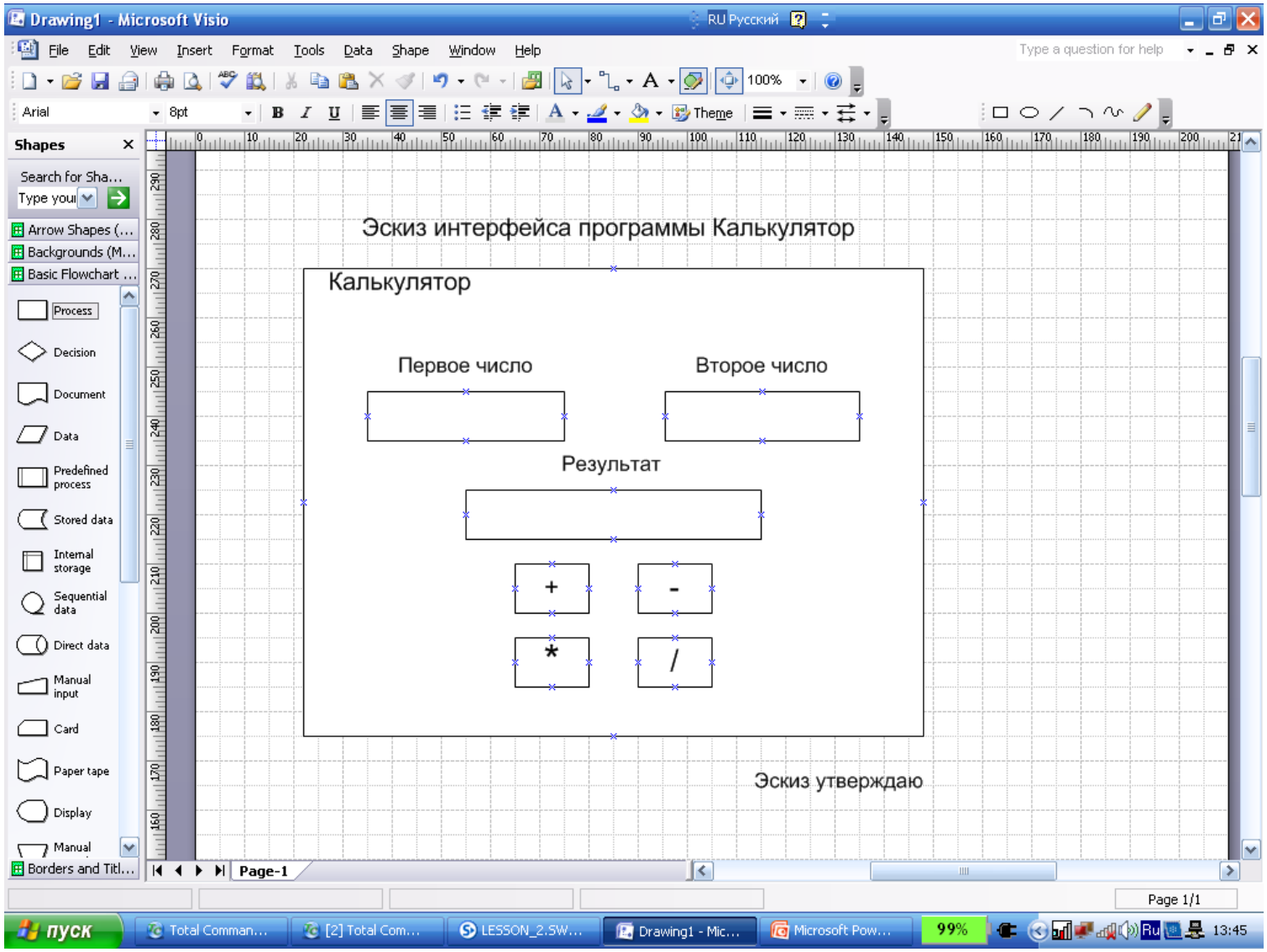


ПРИМЕР 1.

Разработать калькулятор, выполняющий четыре арифметических действия с целыми числами

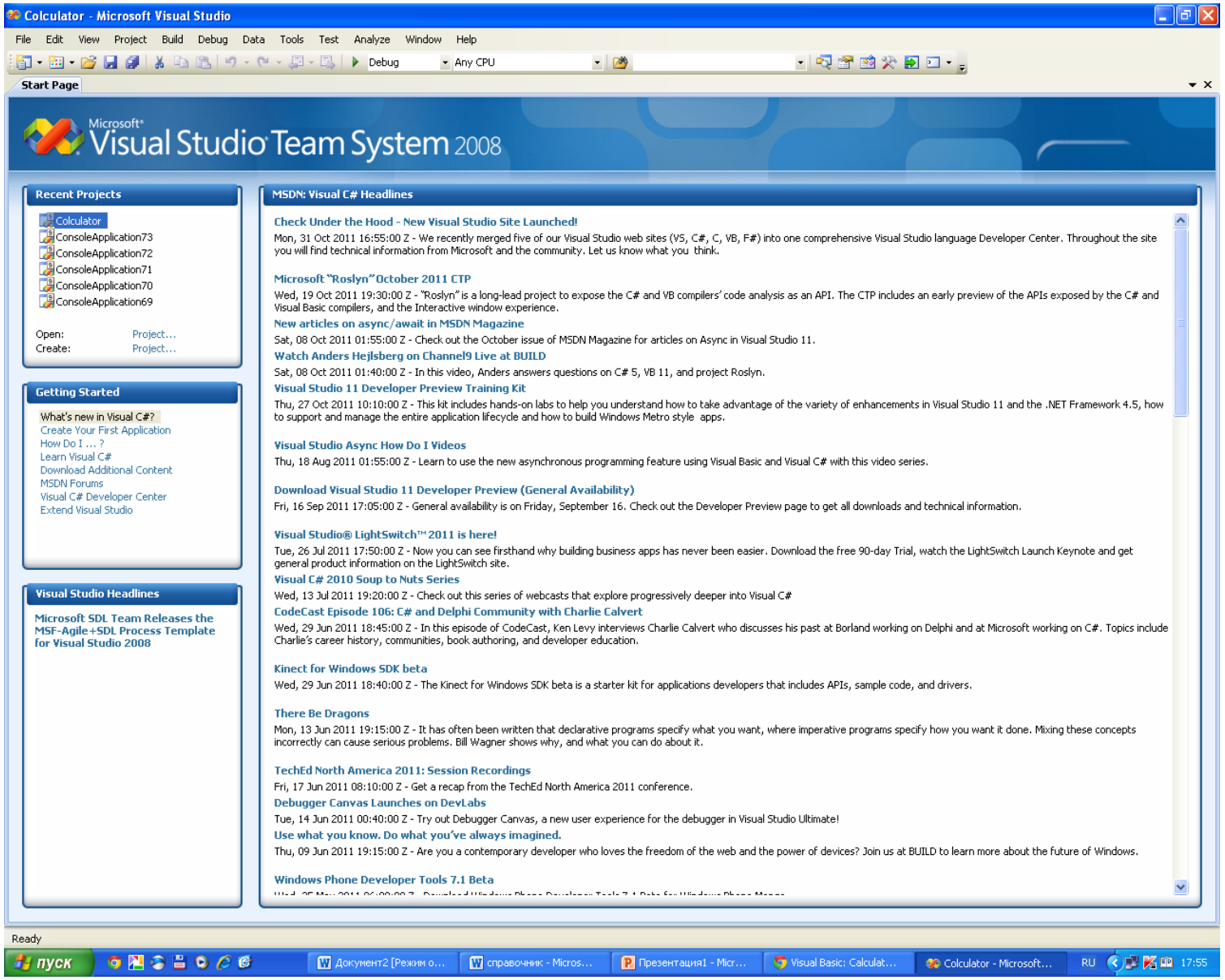
Использовать контролы группы Common Controls: **TextBox, Label, Button**

Разрабатываем эскиз интерфейса приложения в Visio

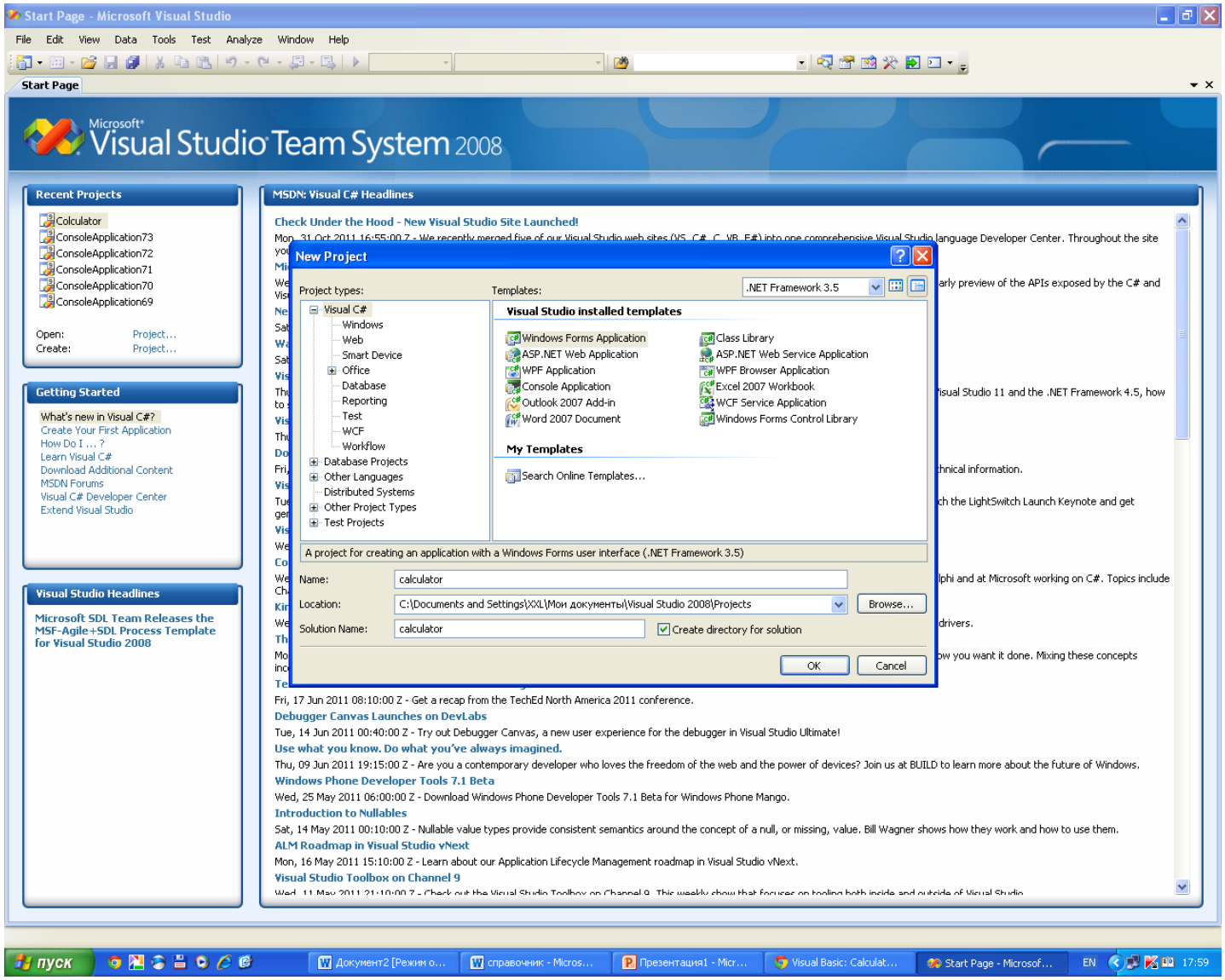


Эскиз рисуем средствами Visio и утверждаем у заказчика.

Загружаем Visual Studio

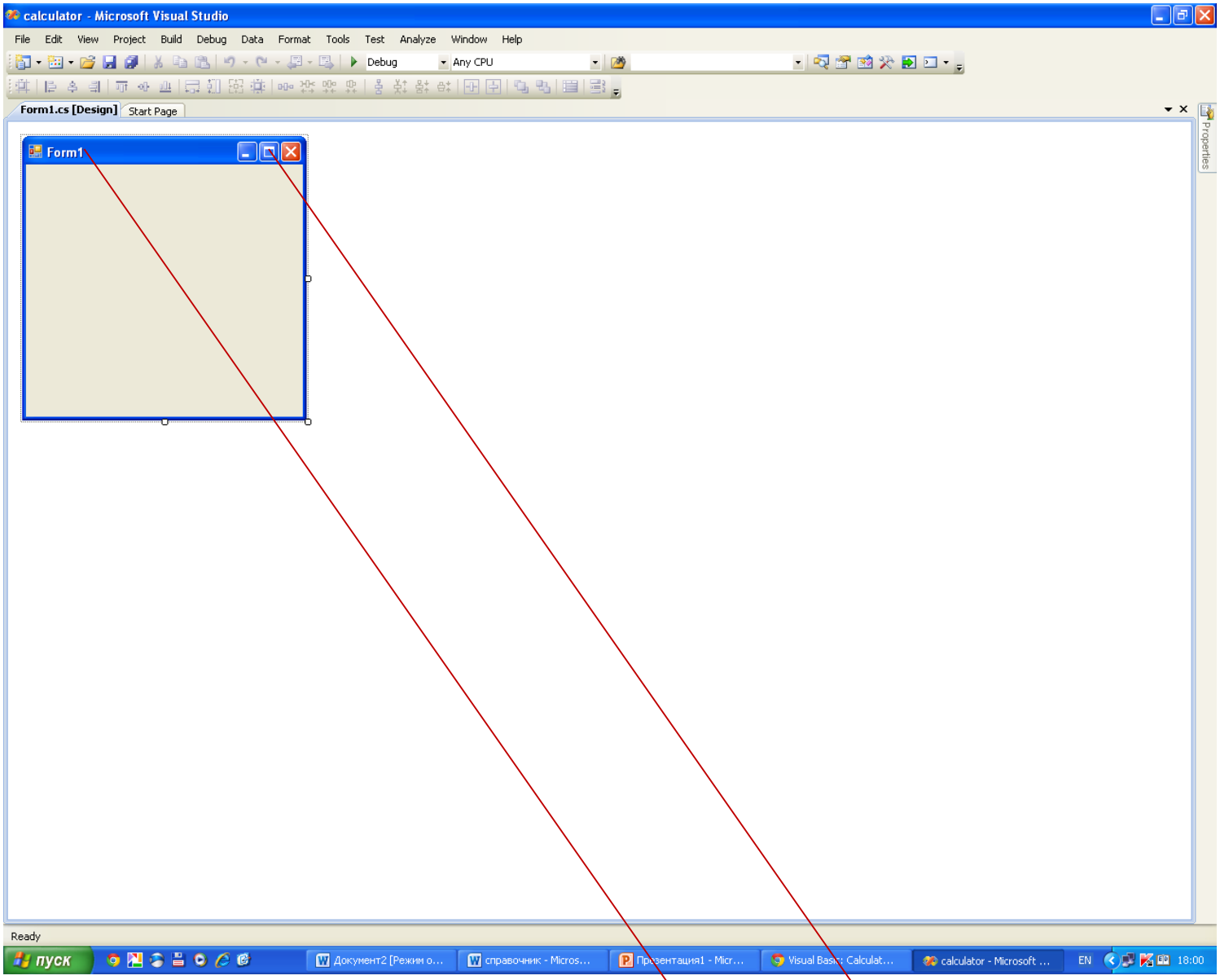


Начинаем новый проект в Windows Forms



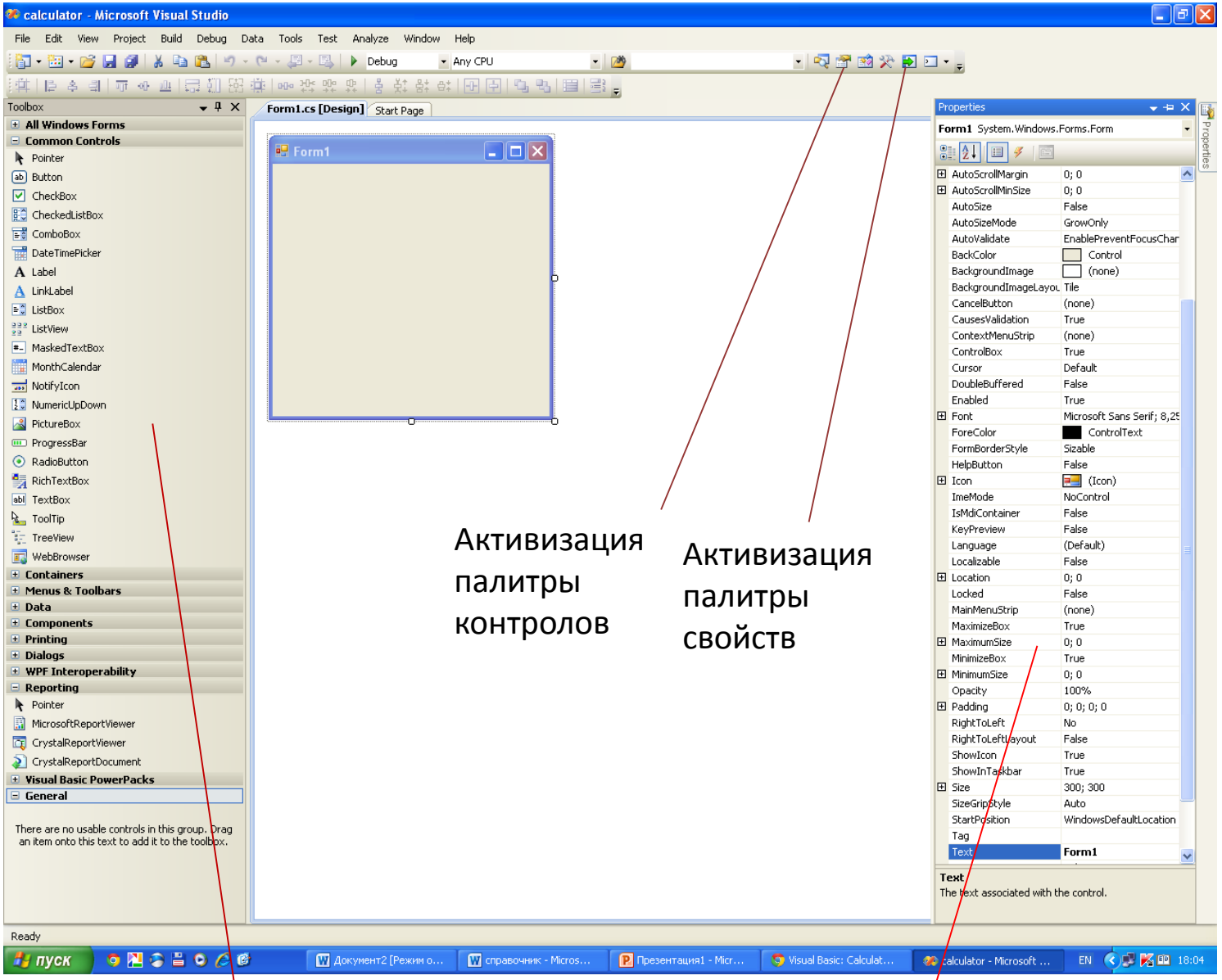
File – New - Project – Windows Forms Application – Name - <имя> (Calculator) - OK

Появляется первичная форма будущего интерфейса

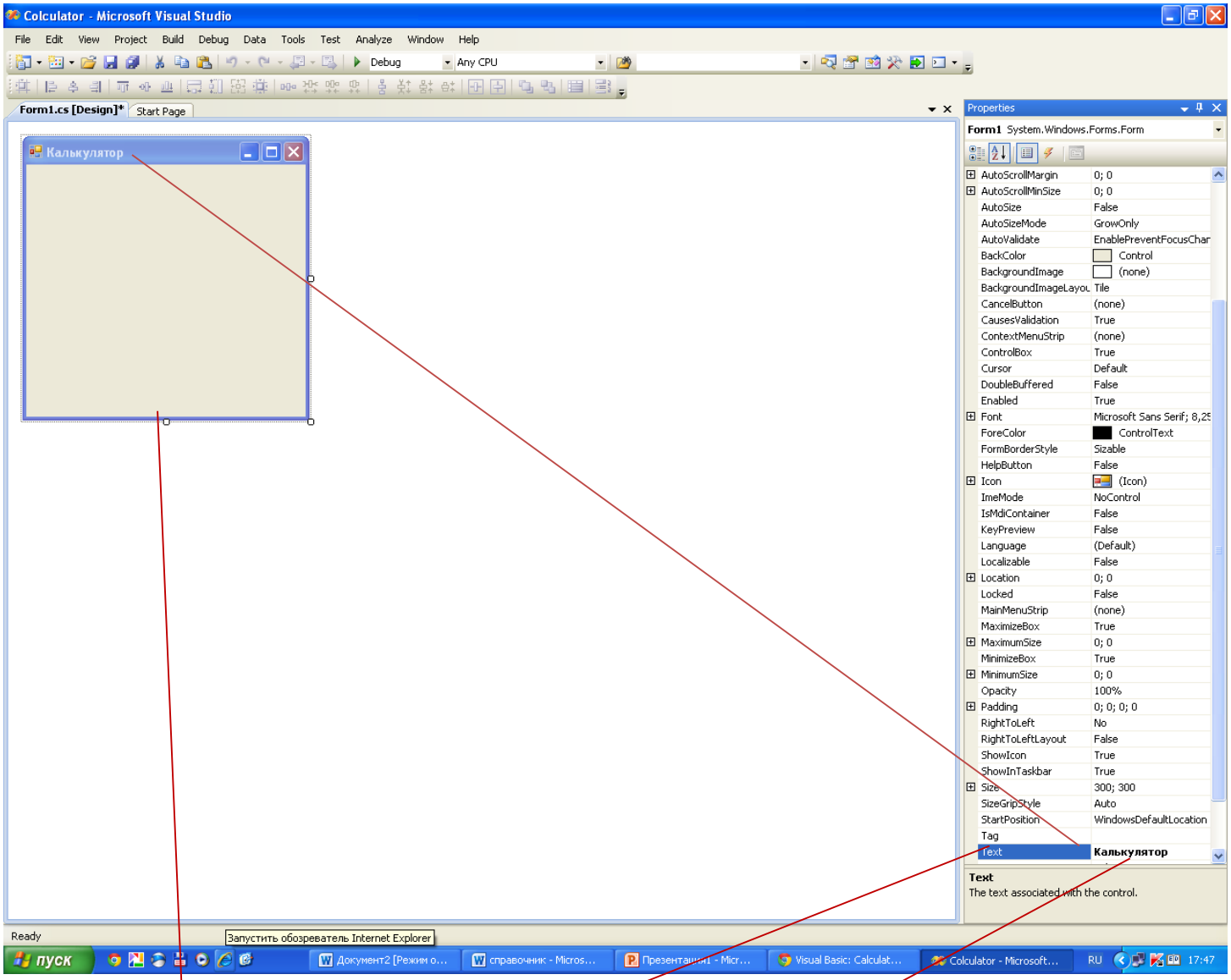


Форма выглядит просто, но уже имеет имя и элементы управления.

Активизируем палитры контролов и их свойств



Меняем название формы: Form 1 на Калькулятор



Форма в фокусе – свойства – **Text** – новое название формы - Калькулятор

Выбираем контролы в Common Controls

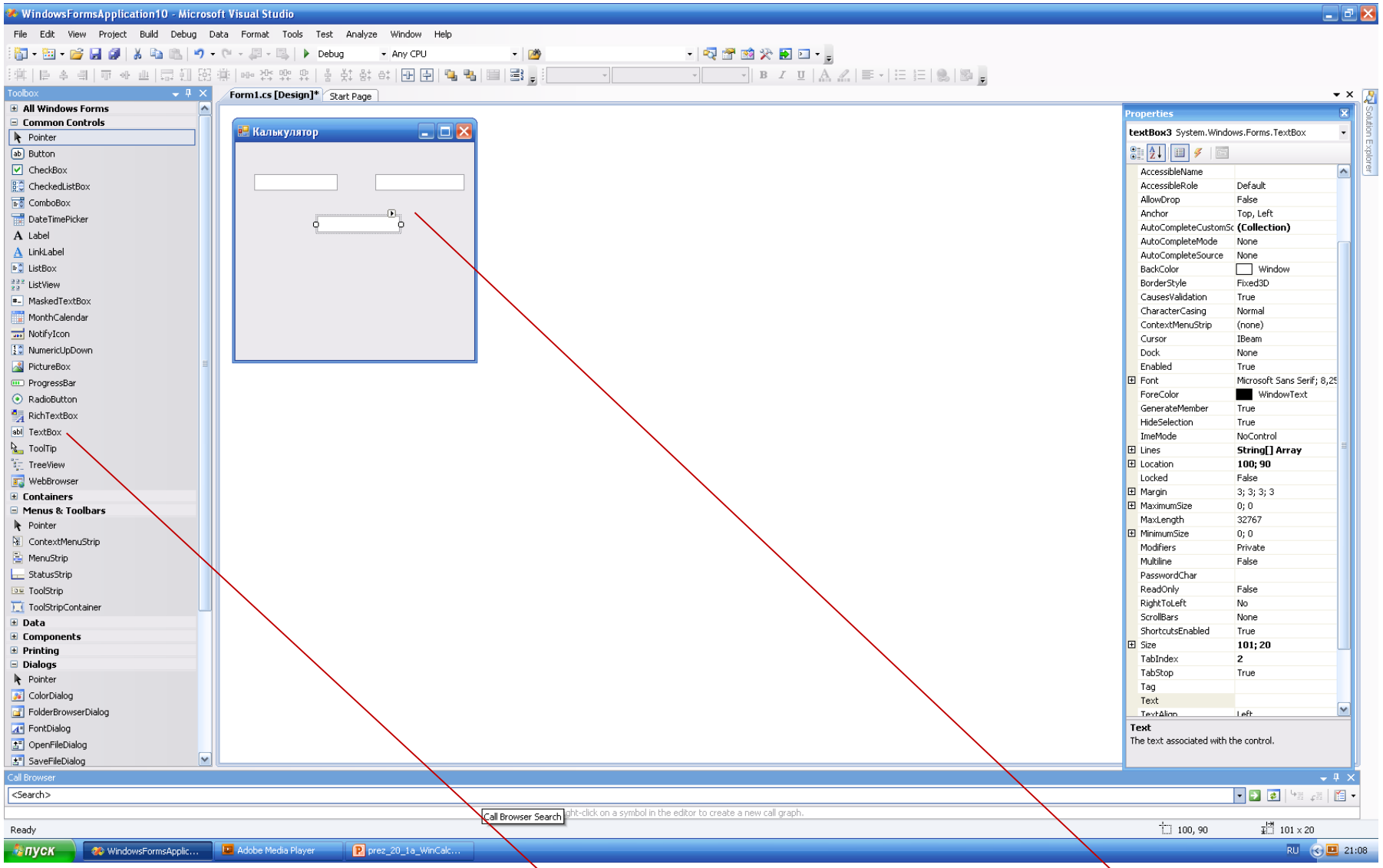
Согласно эскизу, нам нужны контролы:

Button – кнопки, соответствующие операциям +, -, *, /

Label – для описания на форме назначения каждого из окошек TextBox

TextBox – для окошек ввода двух чисел и вывода результата

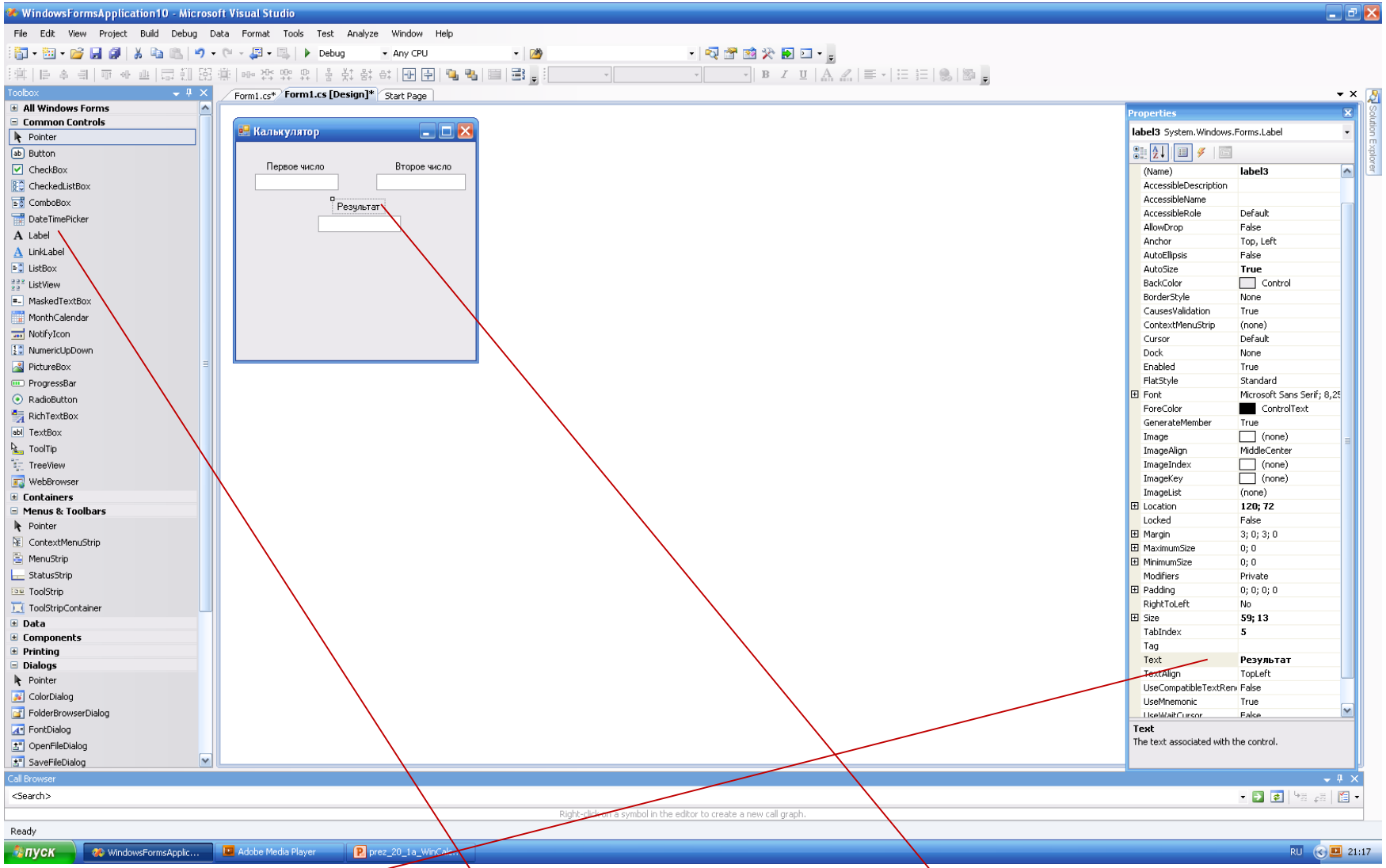
Формируем окна для ввода и вывода



Ищем в списке контролов **TextBox** и перетаскиваем на форму 3 окошка

Вальвачев А.Н.

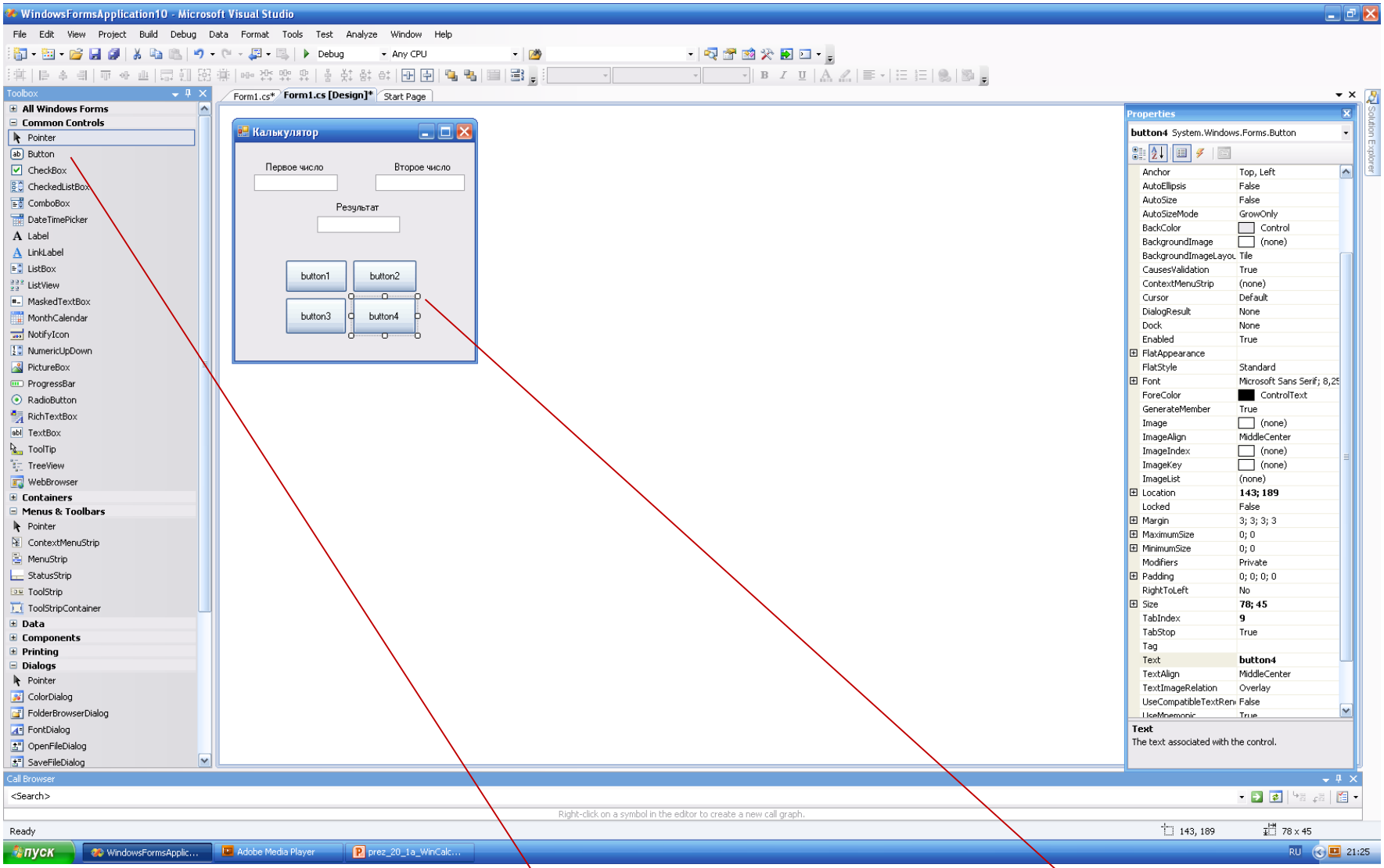
Описываем назначение каждого окошка



Ищем в списке контролов **Label** - перетаскиваем на форму - с помощью **Text** выполняем описание каждого из трех окошек.

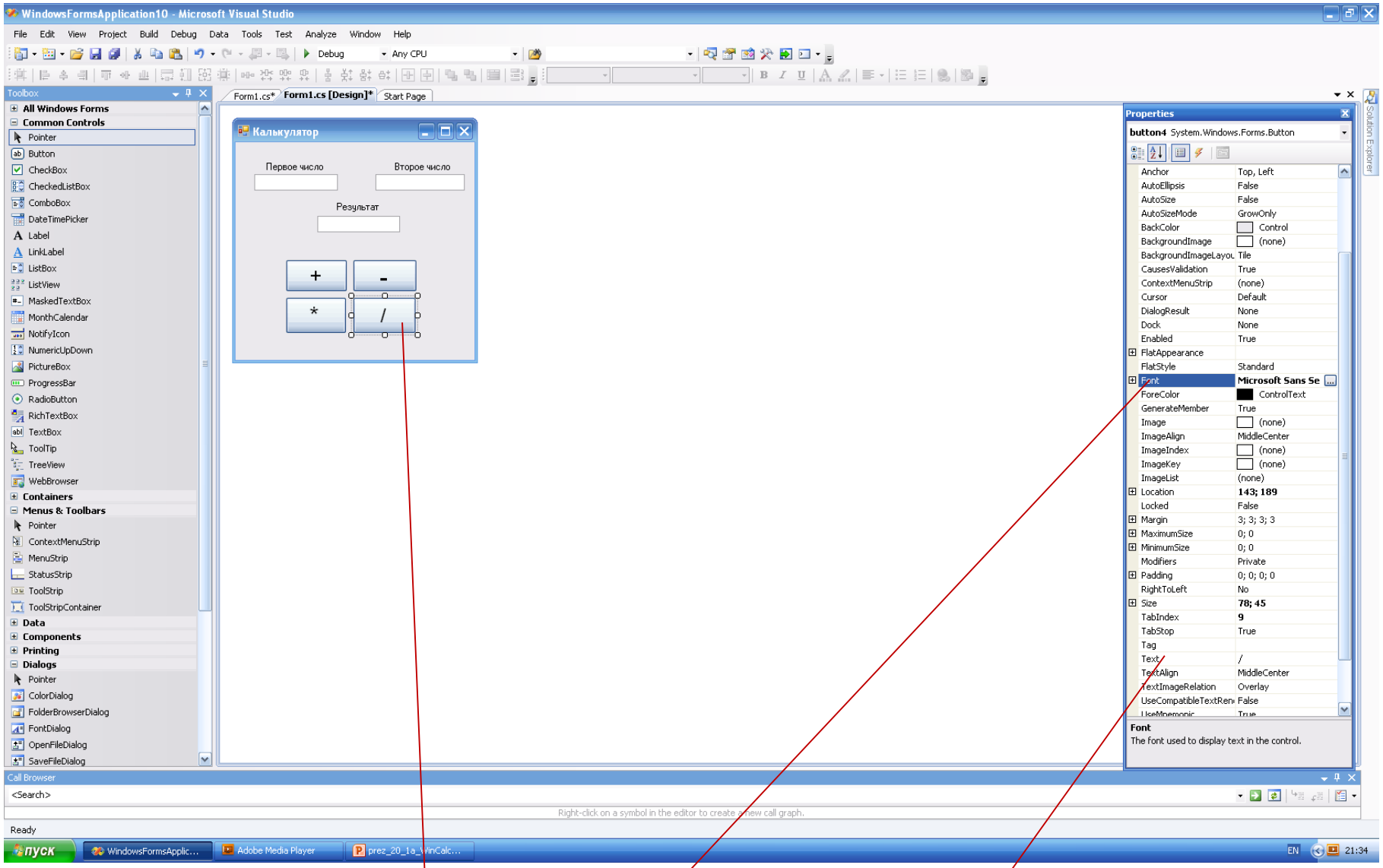
Вальвачев А.Н.

Рисуем четыре кнопки для выполнения операций



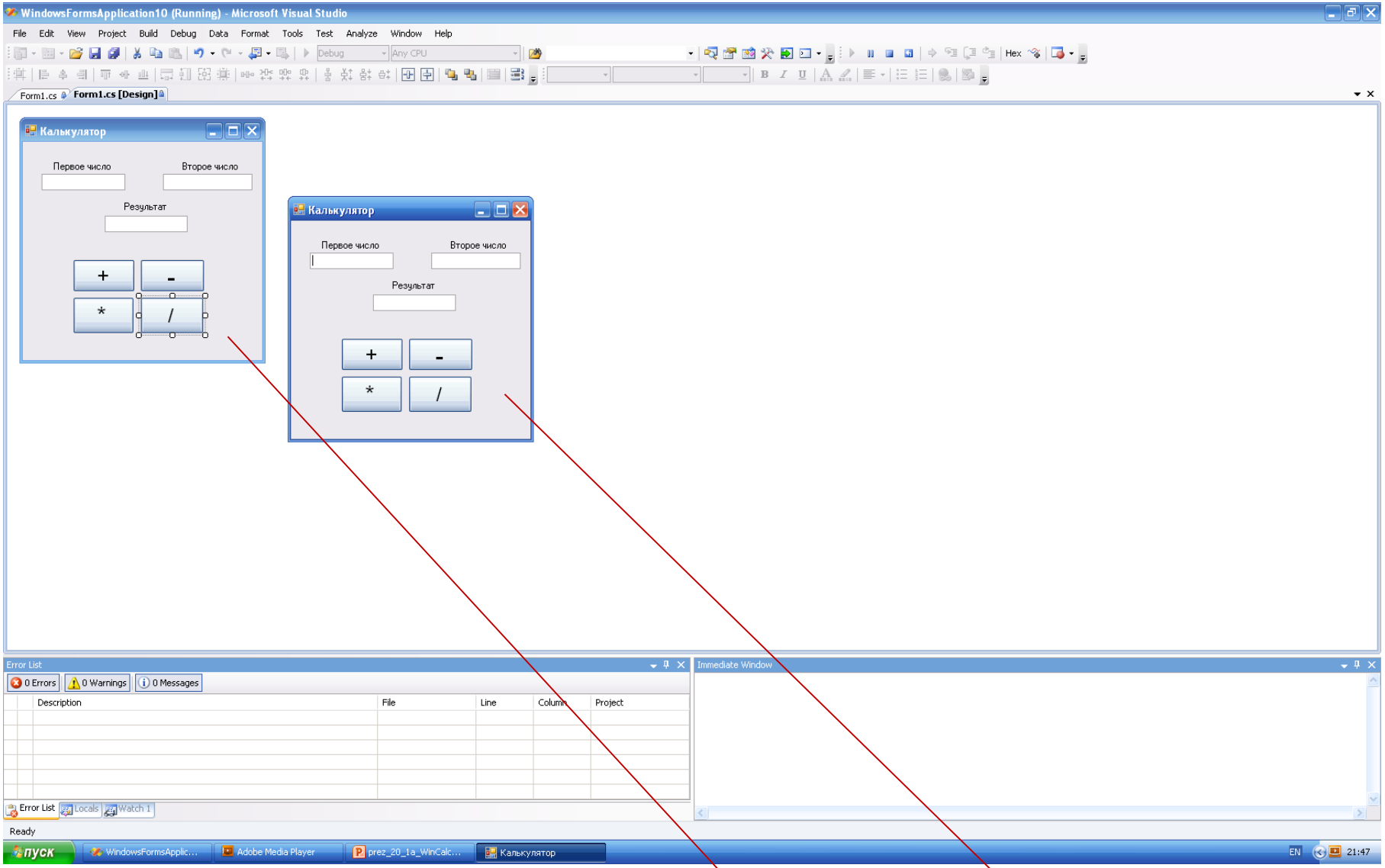
Ищем в списке контролов **Button** и перетаскиваем на форму 4 кнопки

Рисуем на кнопках символы операций + - * /



По очереди активизируем кнопки – выбираем свойство **Text** – вносим символ – корректируем размер символа – **Font**, цвет символа и т.д.

Выполняем пробную компиляцию



Debug – Start Debugging - выводится исходная форма и результат компиляции.

Описываем действия при нажатии кнопок по следующему алгоритму

Этап 1

1. Режим Form1.cs (текст программы)
2. Активизировать первую кнопку
3. Посмотреть и возможно изменить имя кнопки в свойстве Name
4. И так со всеми кнопками

Этап 2

1. Режим Cs.[Design] (графическое изображение формы)
2. Активизировать первую кнопку
3. Режим Form1.cs
4. Выйти в соответствующее место программы
5. Записать операторы реагирование на нажатие кнопки
6. Режим Debug
7. Выполнить пробную компиляцию
8. Исправить ошибки
9. И так со всеми кнопками

Обработка событий при нажатии кнопок:

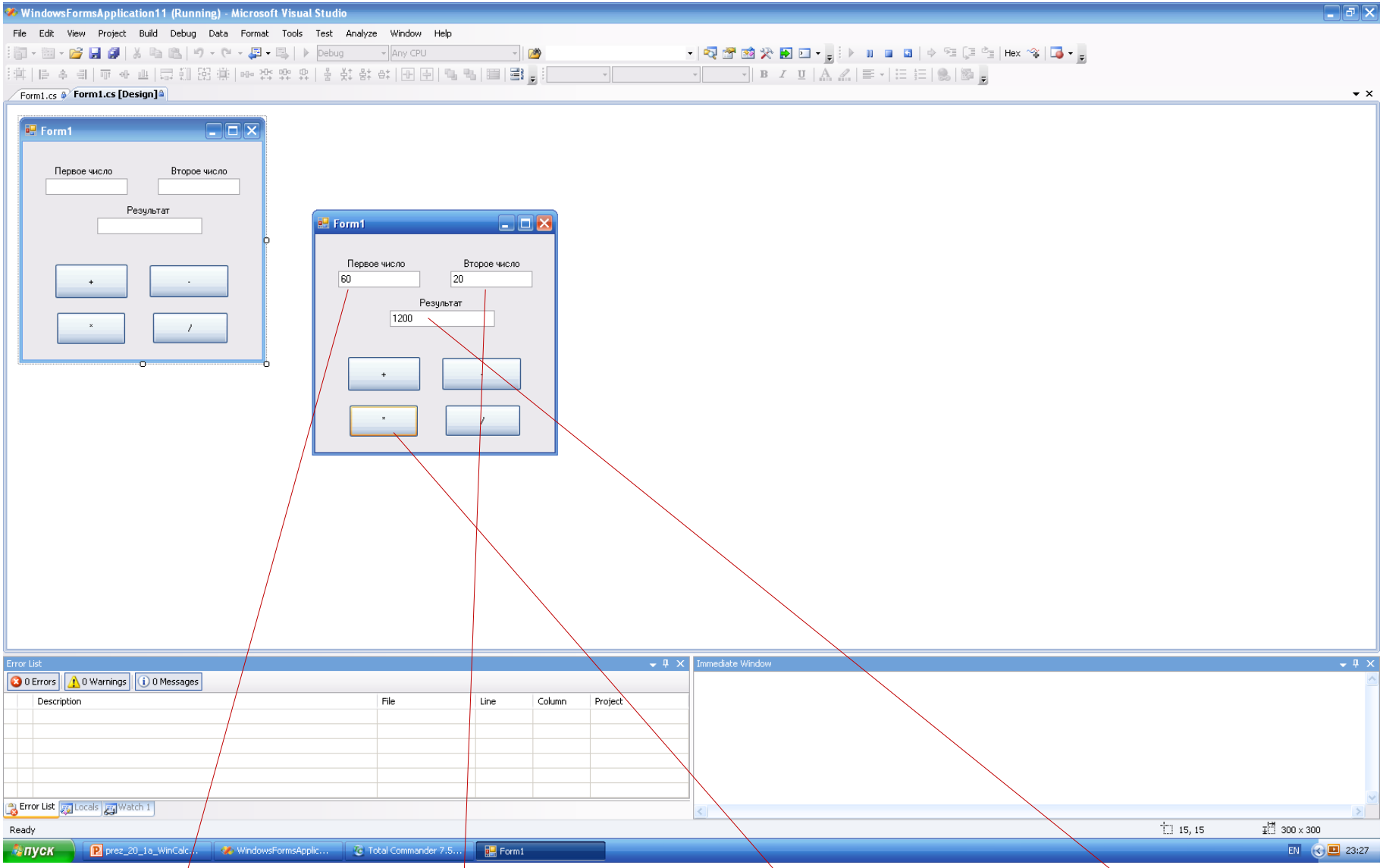
```
 private void button1_Click(object sender, EventArgs e)
{
   textBox3.Text = (Int32.Parse(textBox1.Text) + Int32.Parse(textBox2.Text)).ToString();
}

private void button2_Click(object sender, EventArgs e)
{
   textBox3.Text = (Int32.Parse(textBox1.Text) - Int32.Parse(textBox2.Text)).ToString();
}


private void button3_Click(object sender, EventArgs e)
{
   textBox3.Text = (Int32.Parse(textBox1.Text) * Int32.Parse(textBox2.Text)).ToString();
}

private void button4_Click(object sender, EventArgs e)
{
   textBox3.Text = (Int32.Parse(textBox1.Text) / Int32.Parse(textBox2.Text)).ToString();
}
```

Простой вариант калькулятора работает !



Вводим 60 в первое окно, 20 – во второе, умножаем *, получаем 1200.

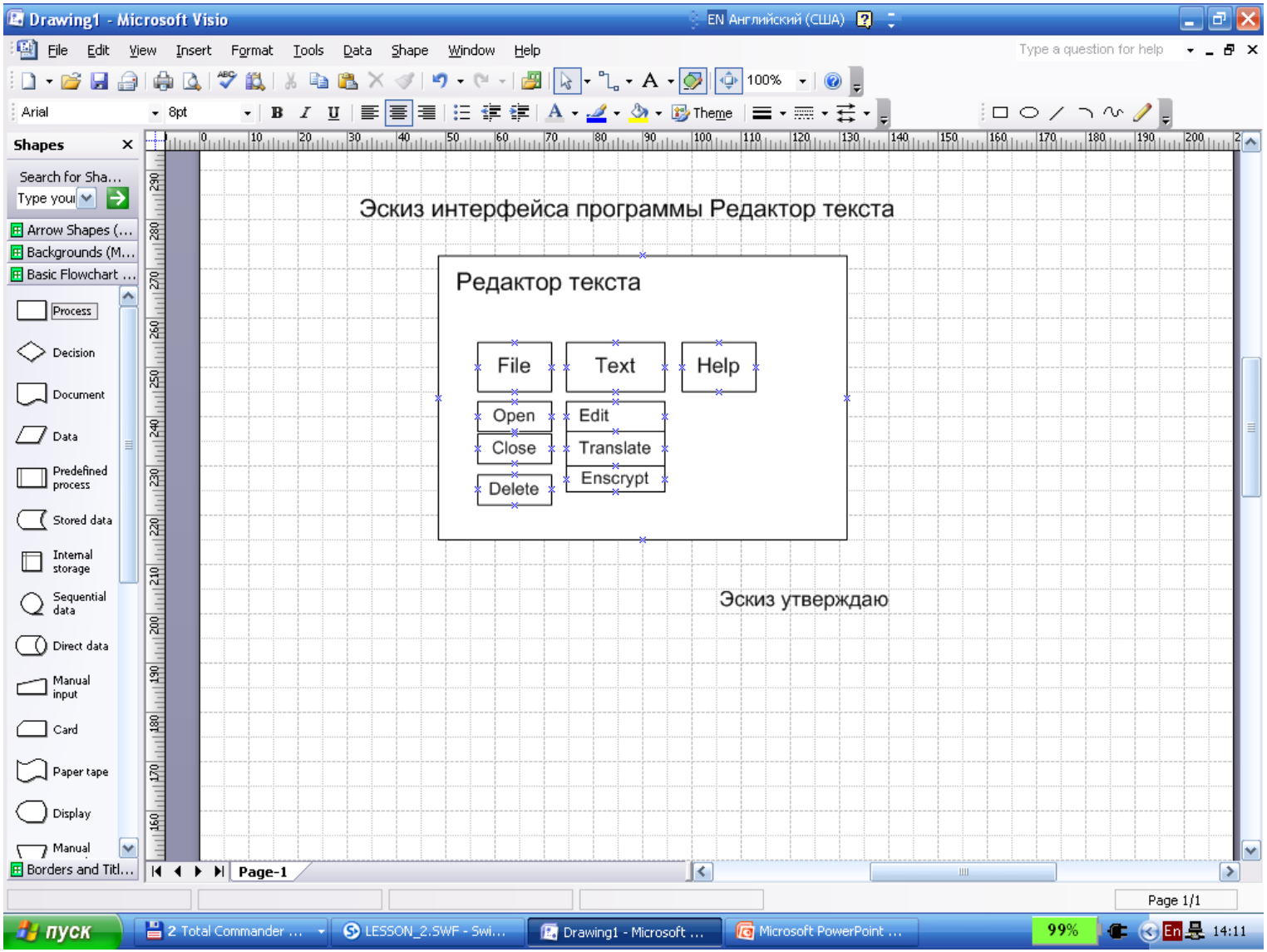
ПРИМЕР 2.

Построить меню для работы с текстовым файлом.

Меню		Операции
File	-	*Open, Close, Delete*
Text	-	*Edit, Translate, Encrypt*
Help		

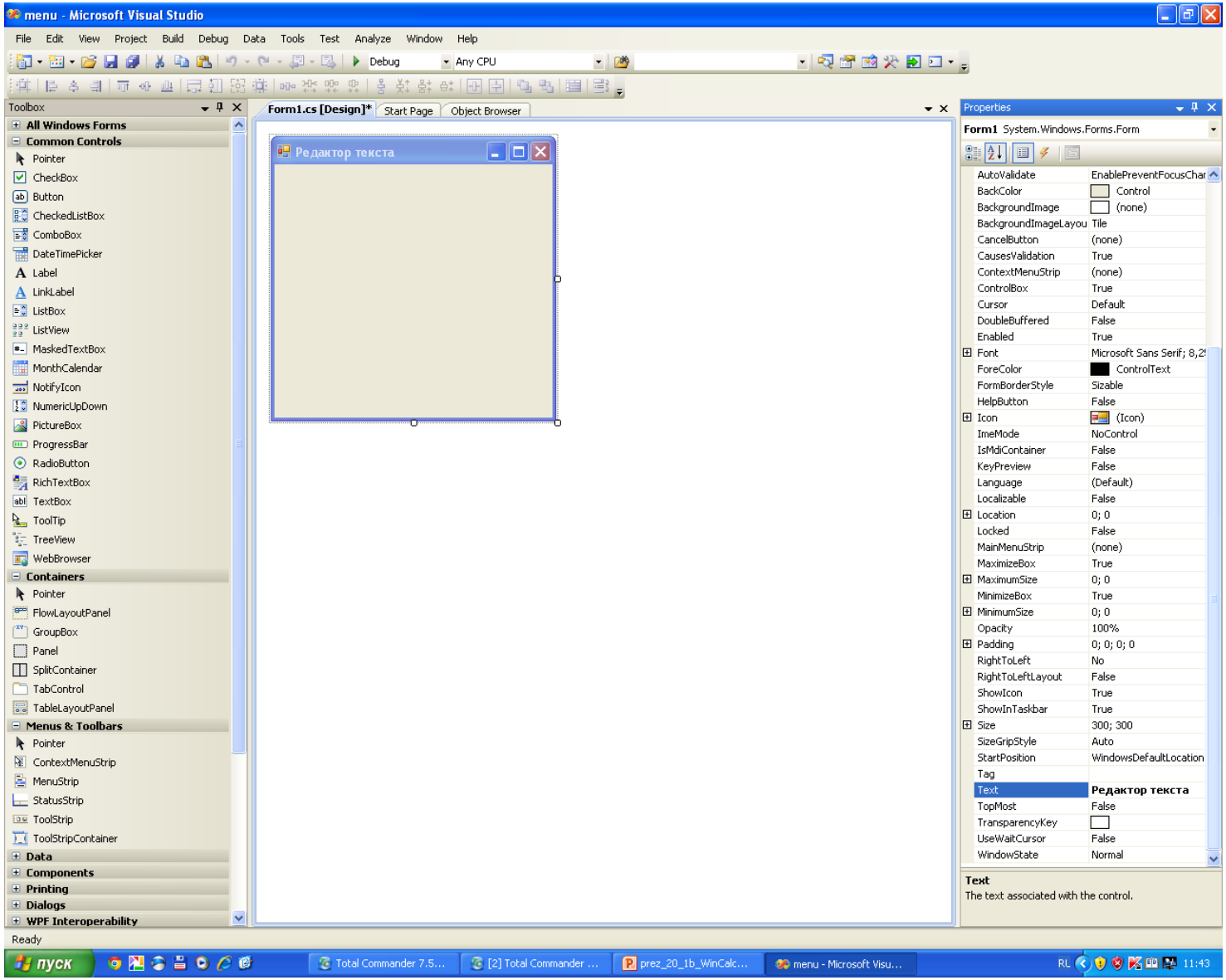
Использовать контрол: **Menus & Toolbars - MenuStrip**

Разрабатываем эскиз интерфейса приложения



Эскиз рисуем средствами Visio и утверждаем у заказчика.

Меняем имя формы

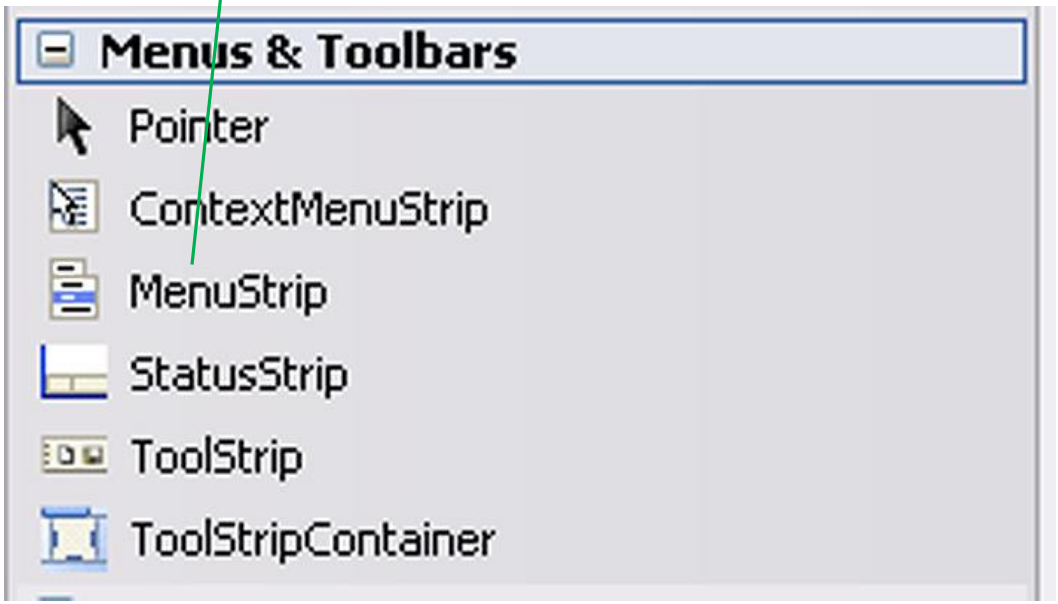


В свойстве Text меняем Form1 на Редактор текста.

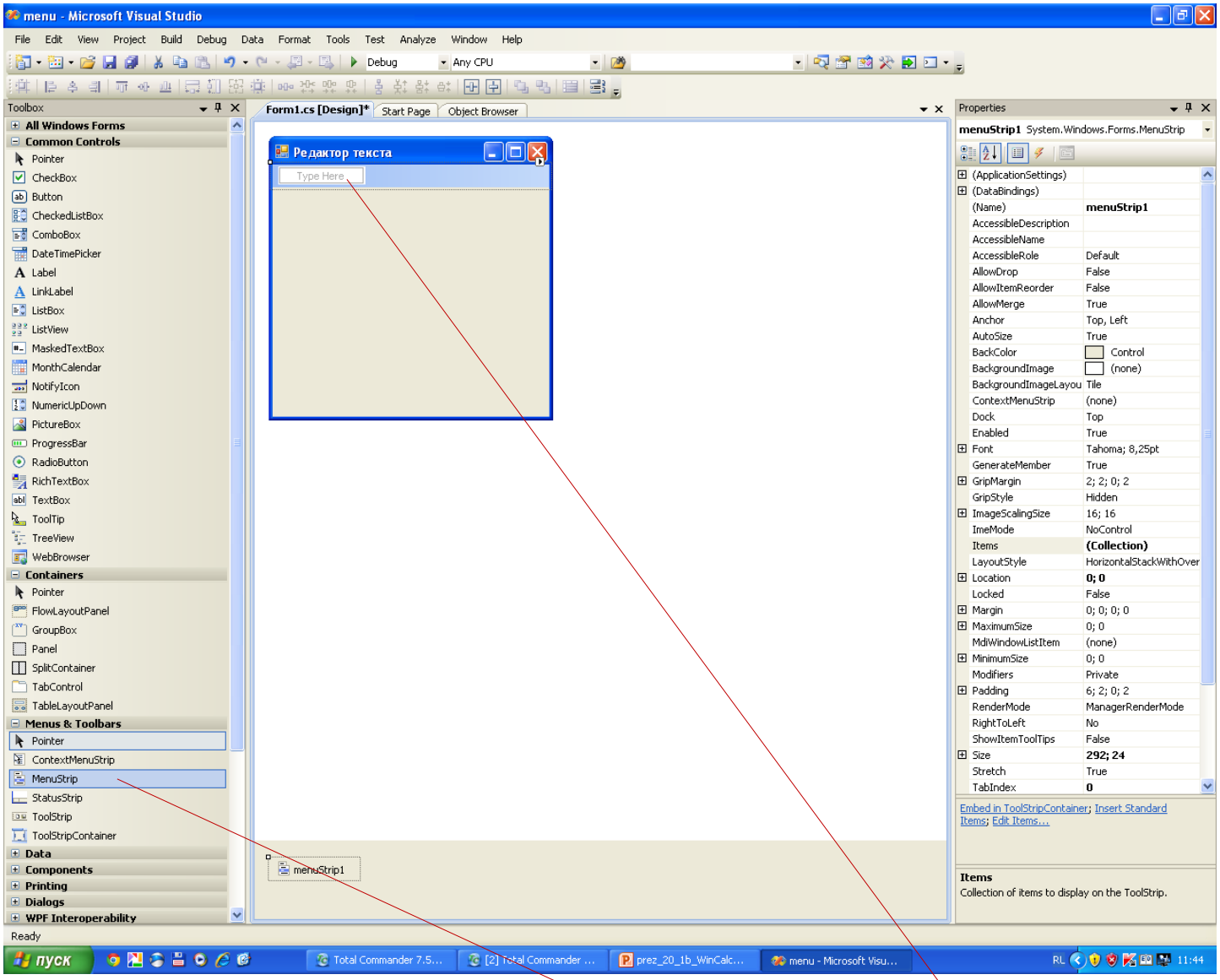
Выбираем контролы

Согласно эскизу, выбираем из палитры контрол:

MemuStrip – из раздела Menus & Toolbars ,
предназначенного для построения меню



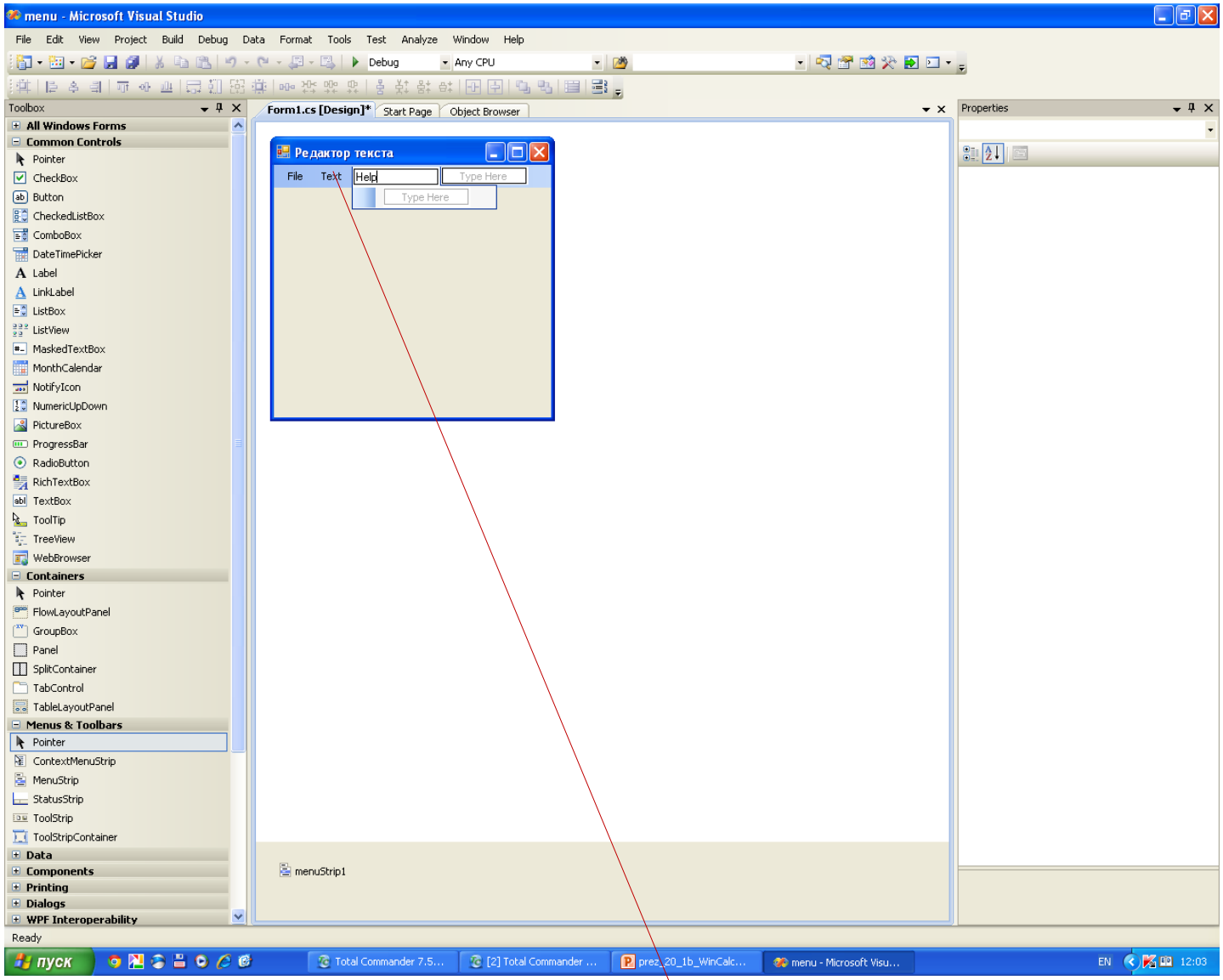
Активизируем контрол построения меню



Клик мышью на контрол **MenuStrip** стартует дизайнер меню.

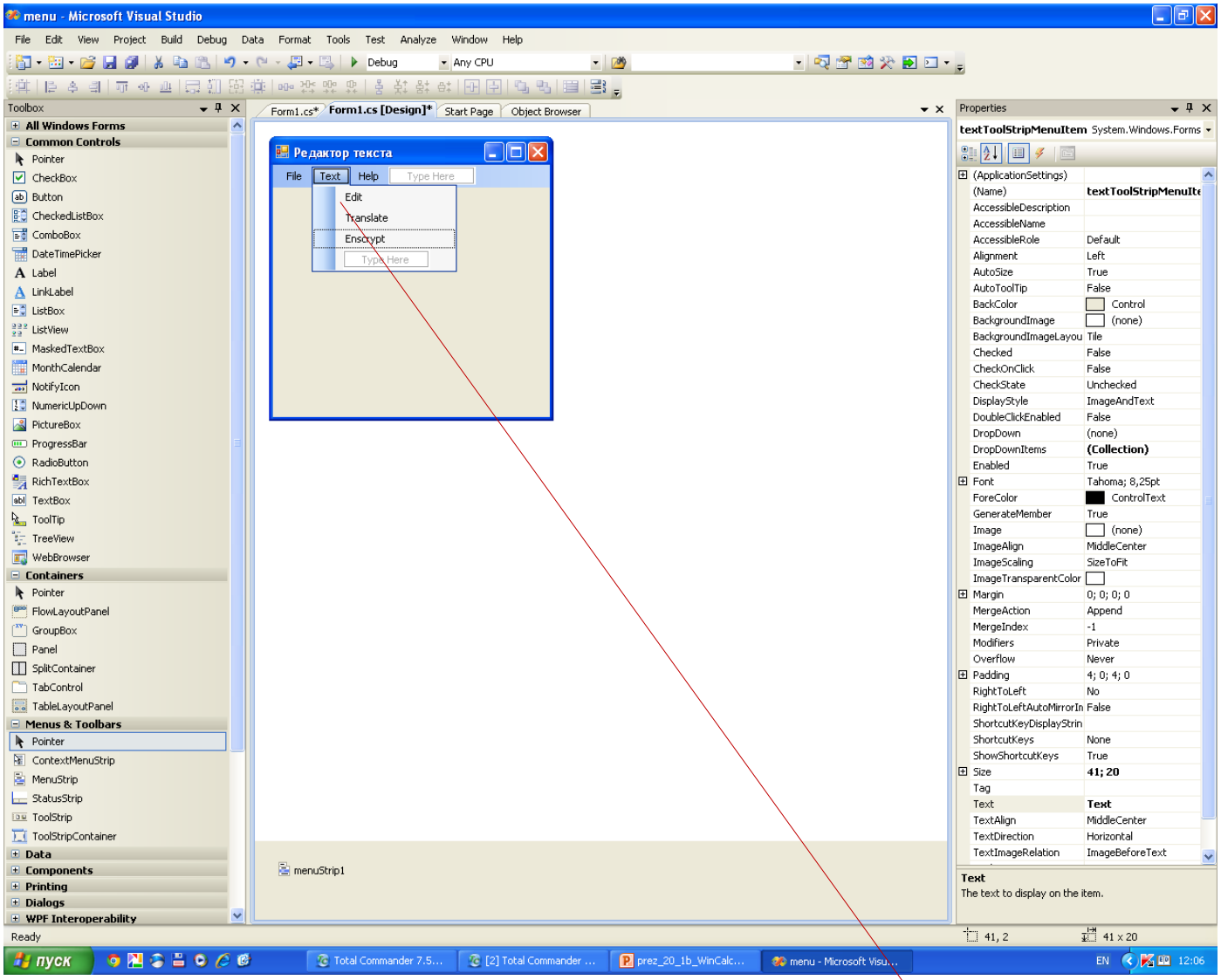
Вальвачев А.Н.

Строим горизонтальные элементы меню



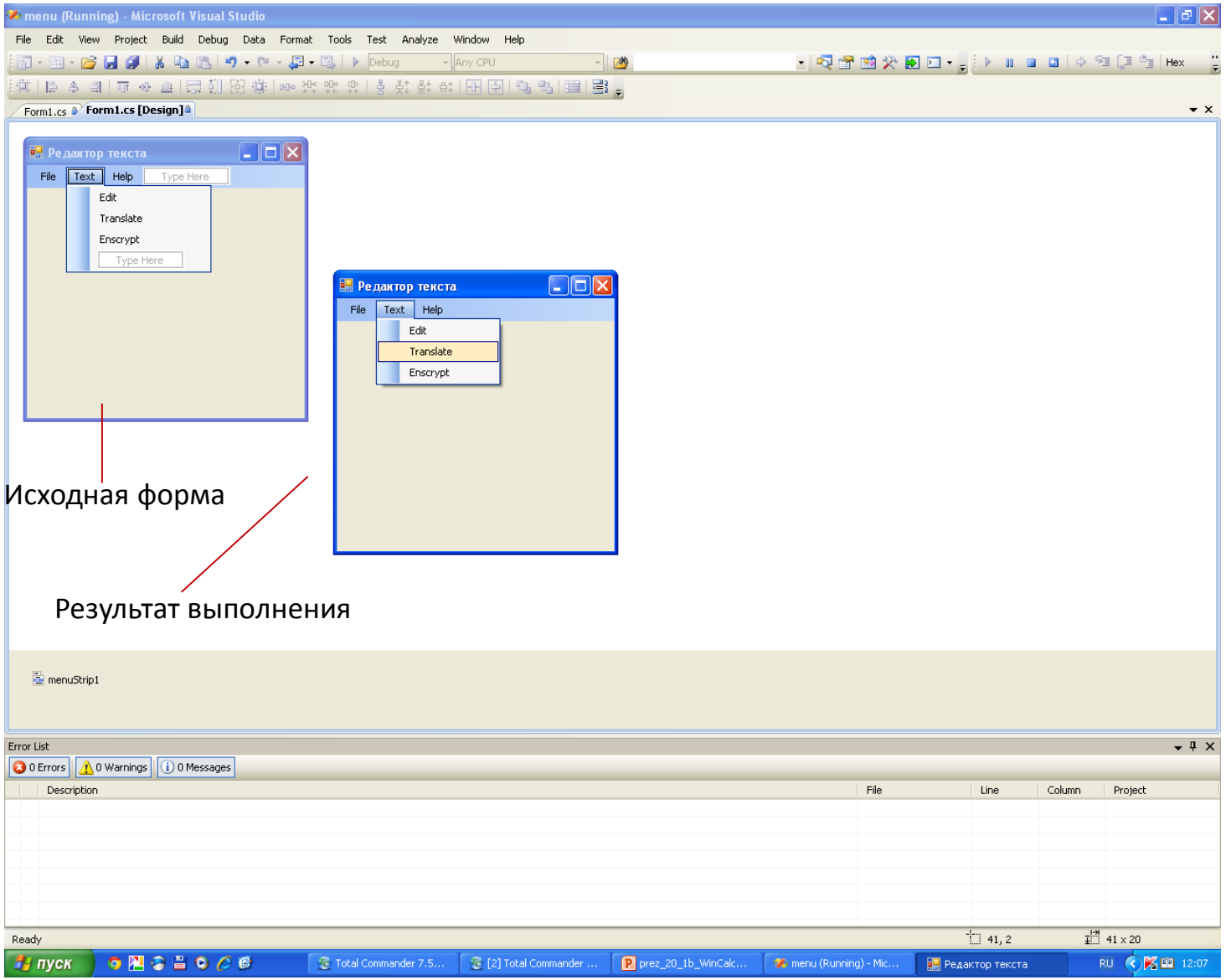
Последовательно добавляем и даем названия элементам меню File, Text, Help.

Строим вертикальные элементы меню



Для каждого горизонтального элемента строим подменю.

Компилируем проект. Меню работает!



Проверяем соответствие элементов меню эскизу и сдаем проект.