**Задание Ш02.** В среде программирования *Microsoft Visual Studio* на языке *Visual C#* разработать простейший графический пользовательский интерфейс с меню. Роль меню заключается в активации возможности начала работы с приложением, а также в предоставлении возможности завершения работы с приложением. Дополнительная роль меню, если таковая предусмотрена, прописана в индивидуальном задании. Роль интерфейсных элементов управления, заданных согласно таблице вариантов индивидуального задания, состоит в изменении индекса, который определяет правила построения графической фигуры в специально предусмотренной для этих целей области для графических изображений (*PictureBox*).

По итогам выполнения работы сдаются строго два файла:

- отчёт, выполненный в текстовом редакторе *Microsoft Office Word* (*\*.doc* или *\*.docx*);

- файл с архивом всего решения (*Solution*), написанного на языке *Visual C#* в среде *Microsoft Visual Studio* в режиме *Windows Forms Application* (*\*.zip* или *\*.rar*).

Отправленные поодиночке файлы проверке не подлежат. При отсутствии одного из упомянутых файлов зачёт по заданию не выставляется.

**Внимание!** Блок-схема алгоритма обязательно должна содержать в себе описание методики построения фигур по координатам и/или сторонам (указываются конкретные точки с их значениями, параметры сторон фигур и прочее в зависимости от реализации). Текст вида «Рисуем окружность», «Пишем текст» и аналогичные записи не являются достаточными. Их обязательно следует дополнить используемыми в программе численными значениями.

Раздел «Подтверждение соответствия графического пользовательского интерфейса требованиям к оформлению» наполнить демонстрацией скриншотами выполнения каждого из пунктов раздела «Требования к оформлению» из данной постановки задачи.

**Требования к именам файлов:**

**Общий вид формата имени файла:** «*Дата. Задание. Фамилия.mcdx*»

**Формат записи даты:** «*ГГГГММДД*», где *ГГГГ* – четыре цифры текущего года, *ММ* – две цифры текущего месяца, *ДД* – две цифры текущего дня.

**Формат записи задания:** «Задание *NNk*», где *NN* – две цифры номера задания, *k* – обозначение «о», если файл содержит общую часть; обозначение «и», если файл содержит индивидуальную часть; обозначение «ои», если файл содержит как общую, так и индивидуальную части.

**Если устранить замечания по работе удаётся в тот же день:** после фамилии ставится пробел и в круглых скобках записывается номер попытки исправления.

**Примеры правильных имён файлов, которые сдаются на проверку впервые:**

«*20190228. Задание 02ш. Иванов.docx*»

«*20190228. Задание 02ш. Иванов.rar*»

«*20190228. Задание 02ш. Иванов.vsdx*»

**Внимание!** Не забудьте выполнить автоматическую нумерацию страниц в отчёте.

Отчёт по выполненной работе должен содержать:

0. Титульный лист.

1. Формулировку цели работы.

2. Описание задачи согласно выданному варианту.

3. Составление диаграммы классов, входящих в состав решения.

4. Составление Сети Петри запрограммированного технологического процесса.

5. Составление блок-схем алгоритмов методов в составе решения.

6. Подбор и расчёт тестовых примеров.

7. Листинг кода составленного программного обеспечения.

8. Графический пользовательский интерфейс программного обеспечения и его описание.

9. Подтверждение соответствия графического пользовательского интерфейса требованиям к оформлению.

10. Расчёт тестовых примеров с использованием составленного программного обеспечения.

11. Формулировку вывода о проделанной работе.

Рекомендации к отчёту, доказывающие самостоятельность выполнения работы и упрощающие процедуру проверки отчёта преподавателем:

1. Выполнение дополнительных скриншотов для случаев, когда текстовое описание проделанных действий становится громоздким или трудным к восприятию.

2. Нумерация рисунков (если есть) с подписями, содержащими названия рисунков, например, «Рисунок 1 – Блок-схема алгоритма программы».

**Цель работы (одна из возможных формулировок)**: «Закрепить навыки разработки визуального пользовательского интерфейса, освоить работу с текстовыми файлами и кодировкой в среде *Microsoft* *Visual Studio*, научиться реализовывать настройку множественных состояний объектов посредством управления компонентами со внутренней индексацией».

**Основное задание на работу**: «В среде объектно-ориентированного программирования *Visual Studio* разработать программу в режиме *Windows Forms Application* на языке *Visual C#*, представляющую собой пользовательский интерфейс, содержащий главное меню, позволяющее:

1. Начать работу с приложением.
2. Прервать работу приложения.
3. Предоставить пользователю справочную информацию о работе с приложением.

Сама программа должна реализовывать вывод в графический элемент (например, *Picture Box*) главной формы плоскостную геометрическую фигуру, выбираемую пользователем из списка (вид списка \* задаётся вариантом индивидуального задания). Список должен обязательно содержать следующие пункты:

1. «Квадрат»,
2. «Прямоугольный треугольник»,
3. «Эллипс»,
4. «Равнобедренный треугольник»,
5. «Круг»,
6. «Равносторонний треугольник»,
7. «Окружность»,
8. «Ромб»,
9. «Трапеция»,
10. «Параллелограмм»,
11. «Прямоугольник».

**Пример Сети Петри для технологического процесса:**

Создадим в *Microsoft Visual Studio* заготовку для программного обеспечения в режиме «*Windows Forms Application*».

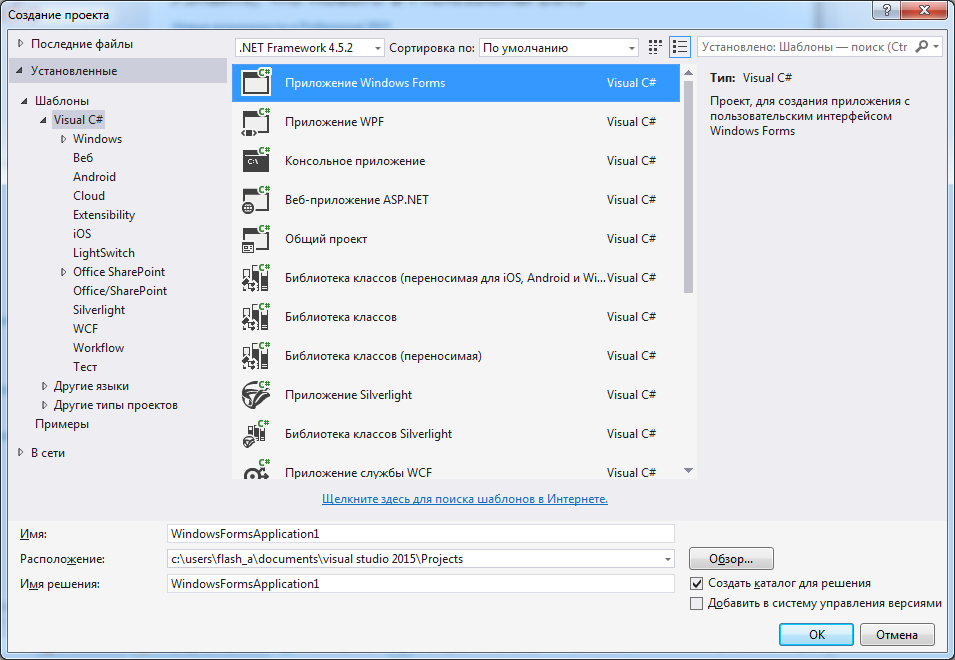


Рисунок 1 – Создание нового решения в режиме «Приложение *Windows Forms*»

В результате выполненных действий на экране в среде программирования будет показана одна единственная форма *Windows* с именем и заголовком «*Form1*».

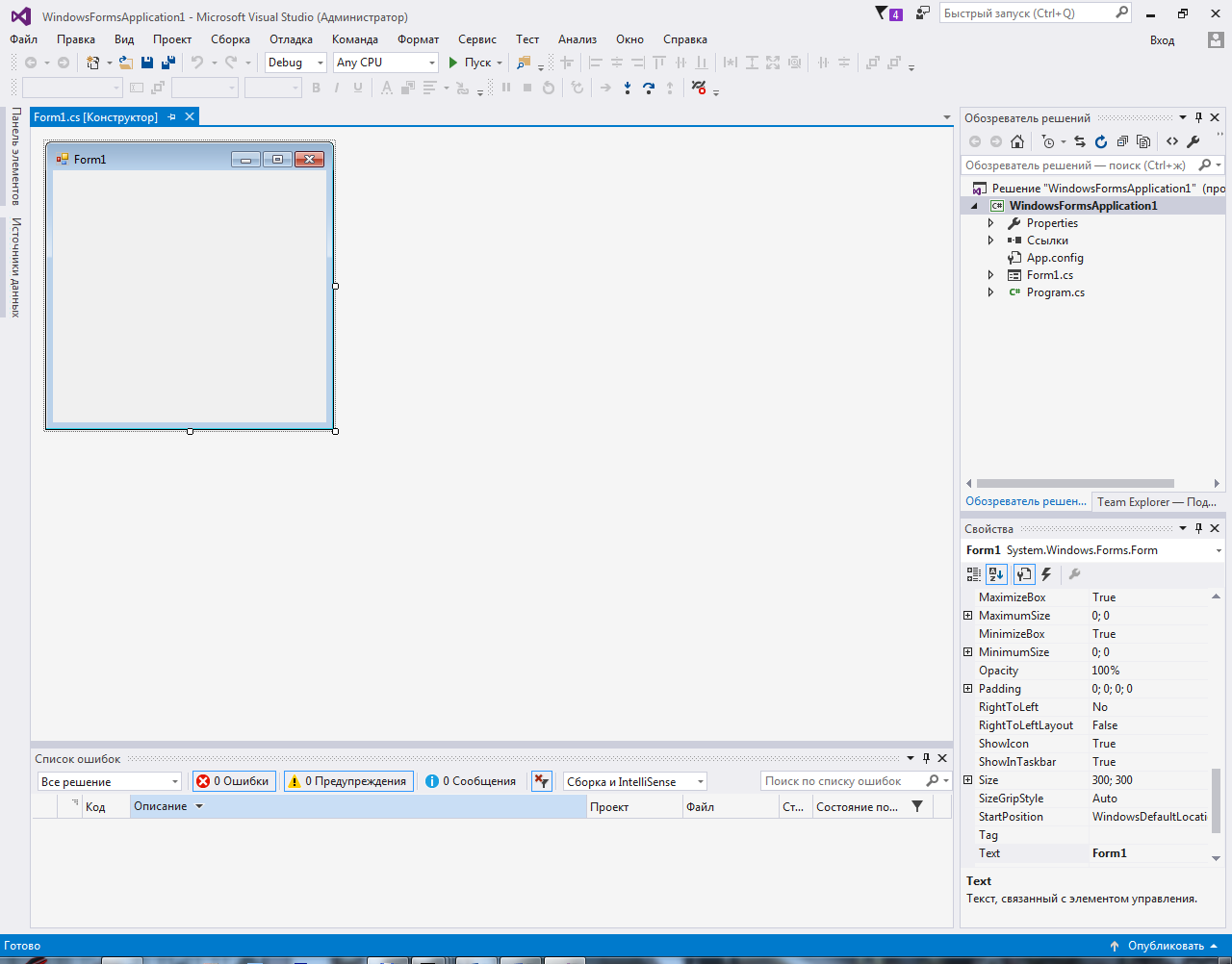


Рисунок 2 – Заготовка приложения *Windows Forms*

Созданный проект можно запустить на исполнение. Он будет скомпилирован и запущен, однако всё, что можно будет сделать в рамках такого проекта с формой это:

- сворачивание / разворачивание формы;

- максимизация / минимизация размеров формы относительно экрана;

- закрытие программы.

Проиллюстрируем технологический процесс взаимодействия пользователя с формой *Windows* сетью Петри. Эта модель позволяет исключить потоковую составляющую из процесса и, практически, полностью исключить время ожидания системой тех или иных действий пользователя.

Начало и конец работы программы иллюстрируются в сети треугольниками, компоненты модели различными фишками подразделяются на события, устойчивые состояния, методы, функции и процедуры. Горизонтальные площадки аккумулируют и группируют совместные события для реализации переходов к определённым действиям или группам дейстивий.

****

Рисунок 1 – Сеть Петри технологического процесса загрузки главной формы решения, не содержащей ни одного дополнительного элемента управления (начало)

****

Рисунок 2 – Сеть Петри технологического процесса загрузки главной формы решения, не содержащей ни одного дополнительного элемента управления (продолжение)

**Пример настройки компонентов через конструктор:**

**Пункты и подпункты меню:**

В панели элементов выберем компонент «Полоса меню» («*Menu Strip*») и разместим его на главной форме созданного решения. Компонент автоматически притянется к верхней границе формы.

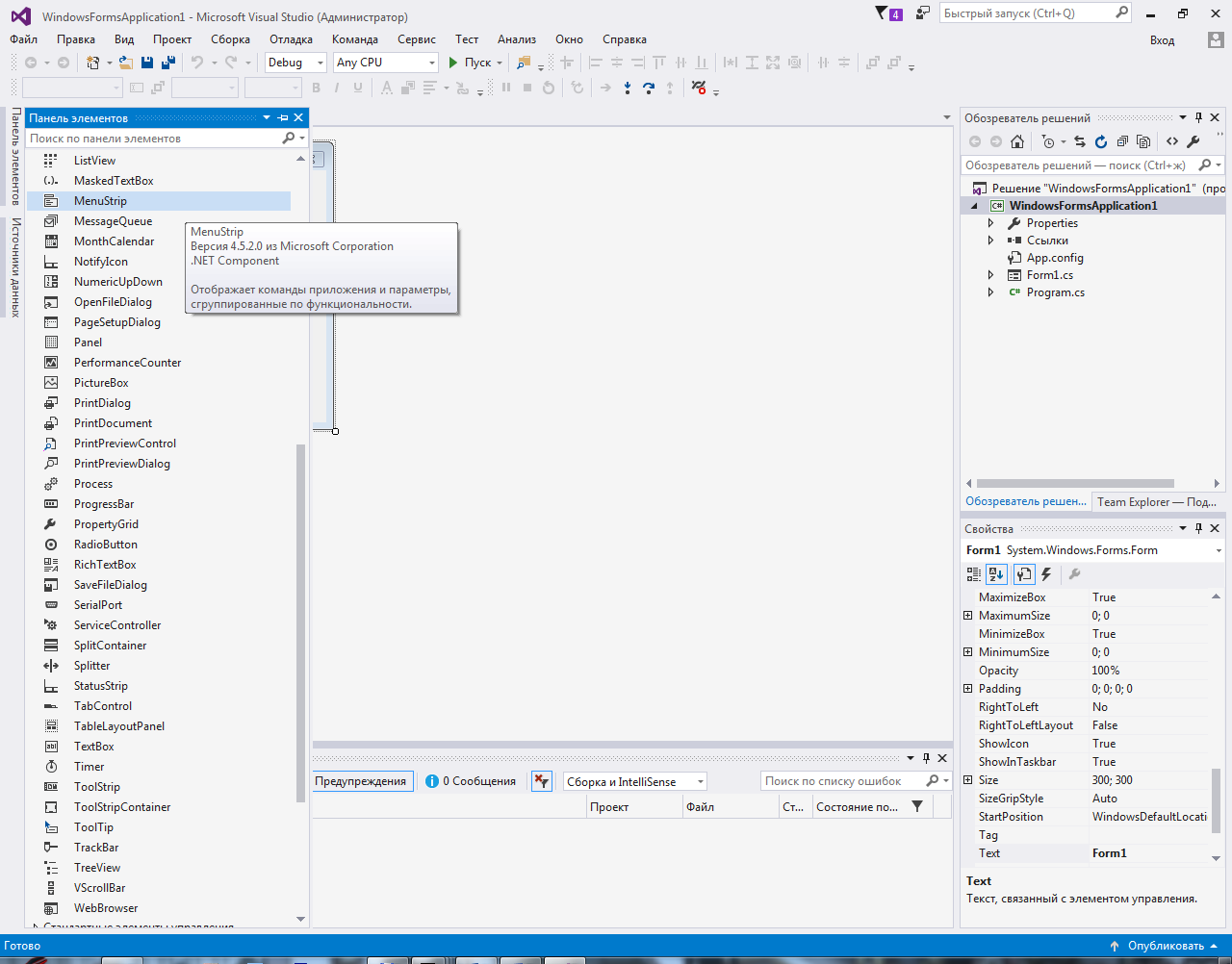


Рисунок 3 – Выбор из перечня компонентов полосы меню

При выборе компонента появляется визуальная подсказка «Вводить здесь», что позволяет наполнять полосу меню содержимым.

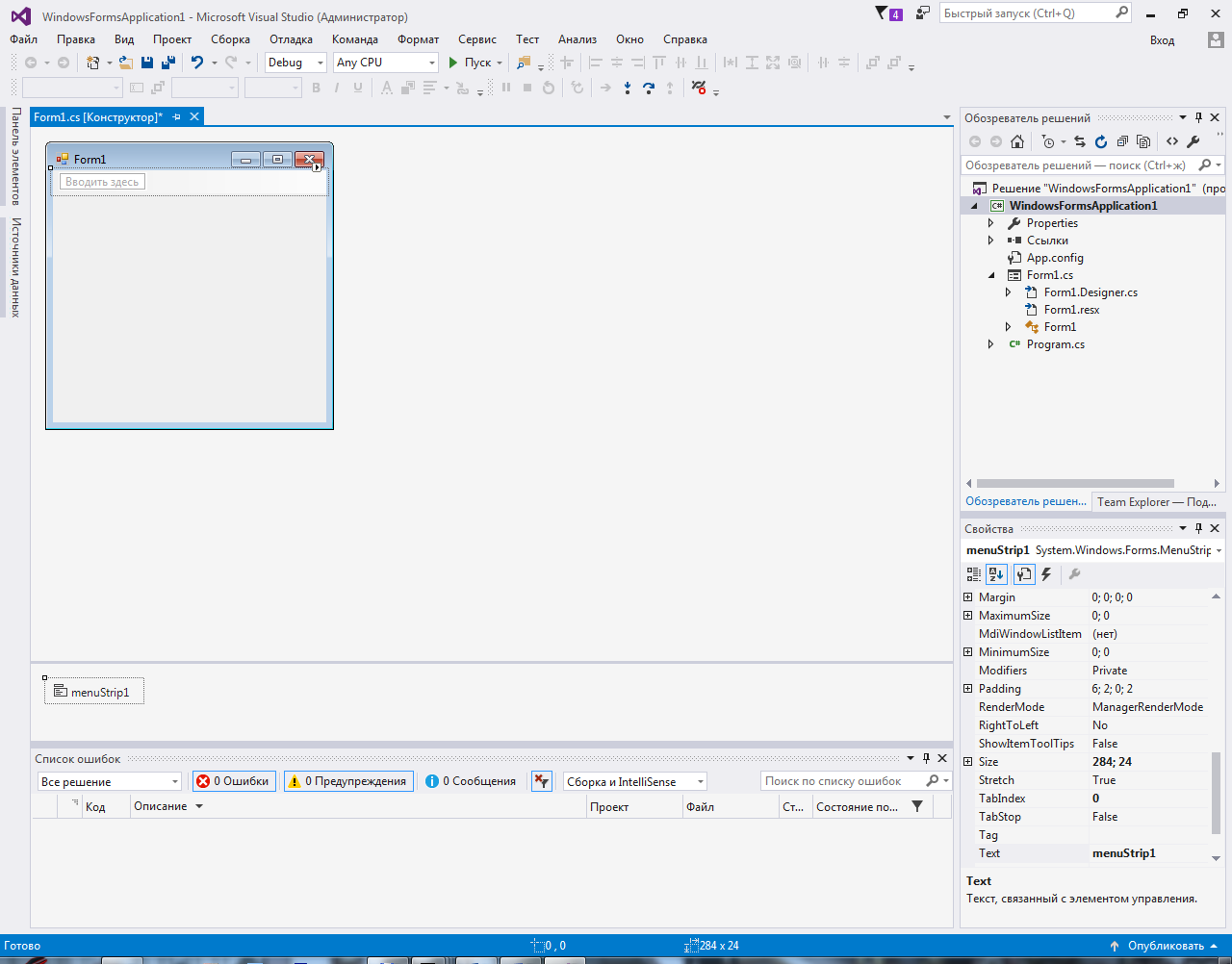


Рисунок 4 – Размещение полосы меню на форме *Windows*

Содержимое можно заполнить вручную или выбрать из выпадающего списка, в котором представлены: «элемент меню» («*MenuItem*»), «комбинированный список» («*ComboBox*»), «текстовое поле» («*TextBox*»).

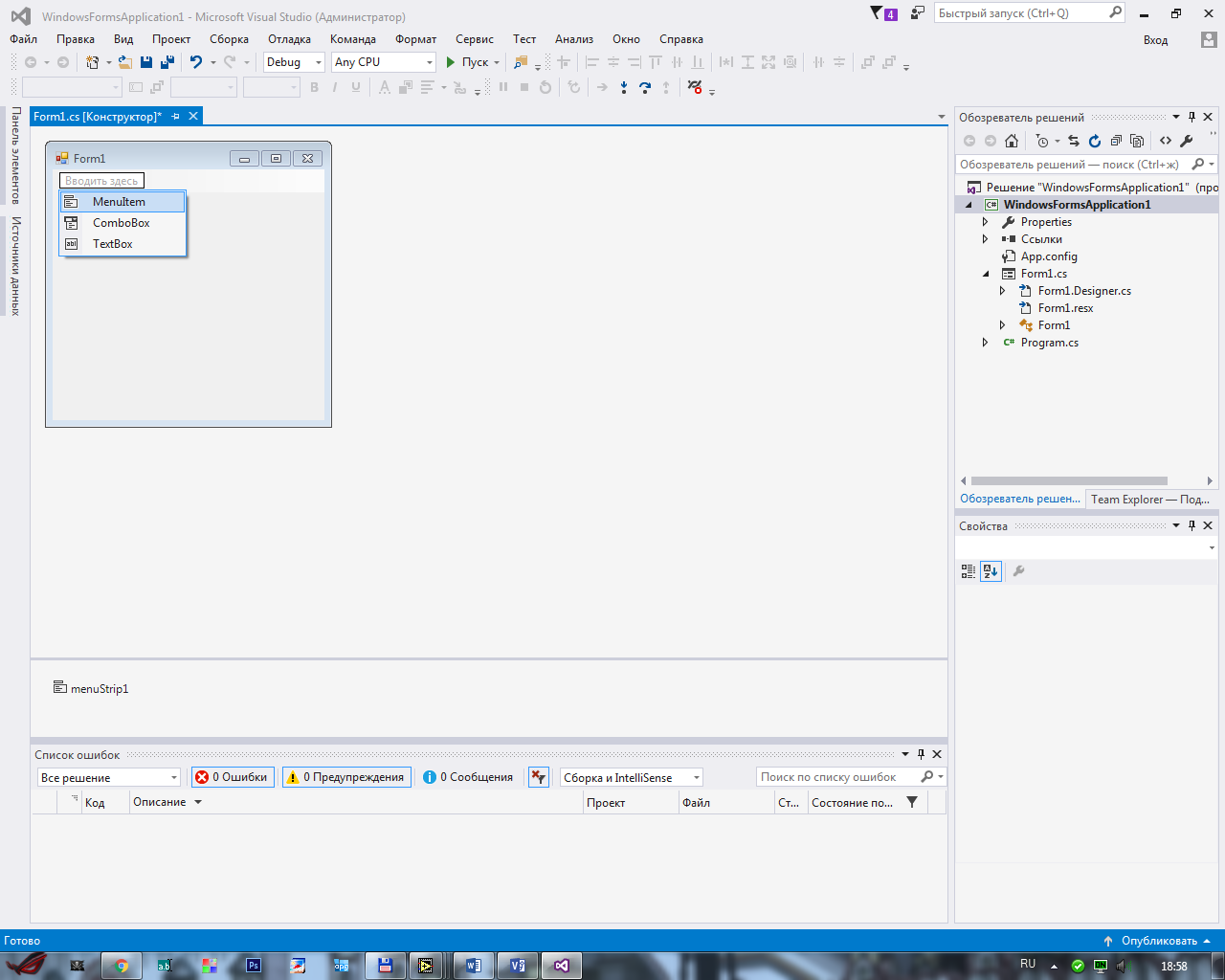


Рисунок 5 – Под каждый пункт меню в конструкторе имеется задел, в котором можно выбрать варианты: пункт меню, комбинированный список, текстовое поле

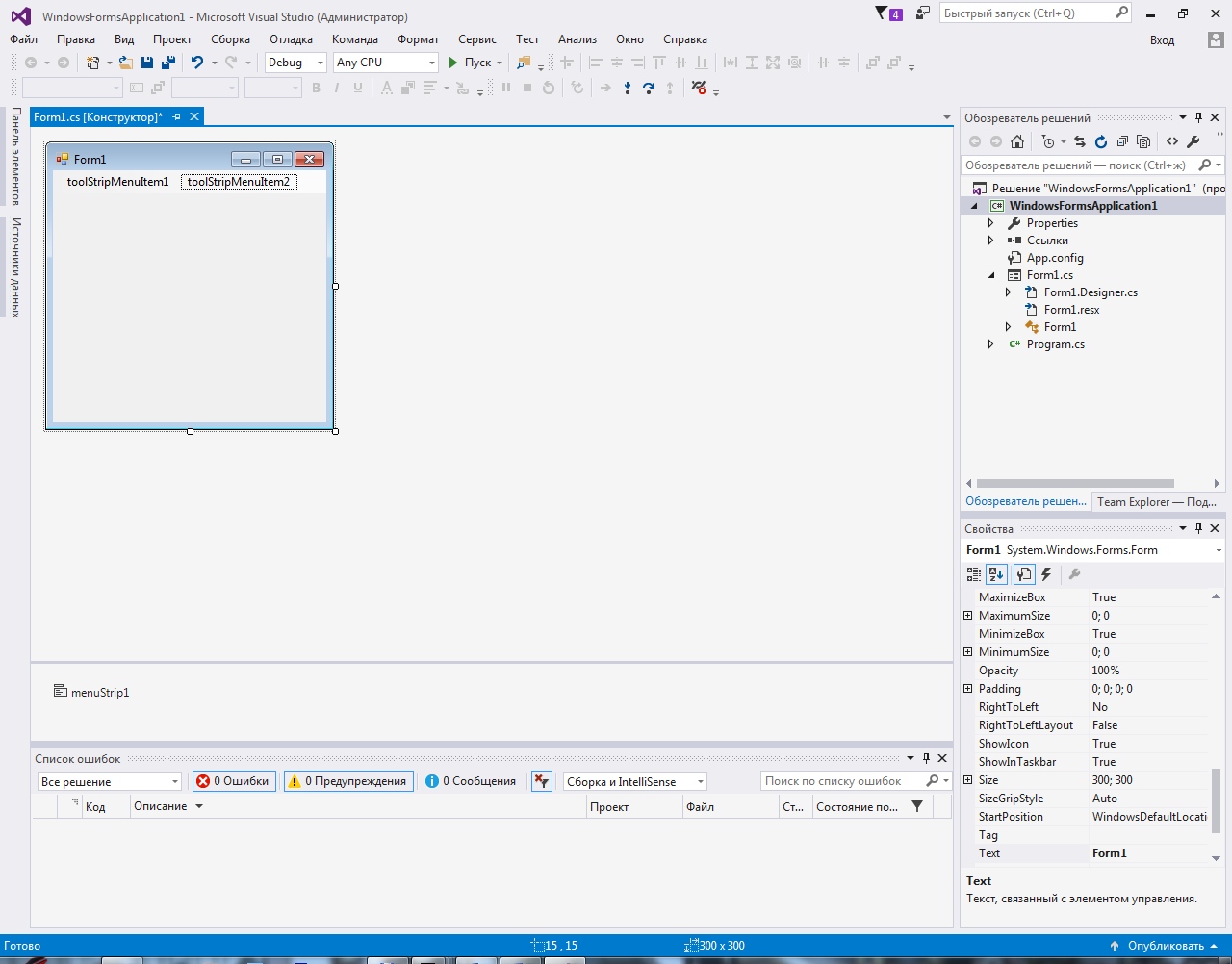


Рисунок 6 – Созданы два пункта меню

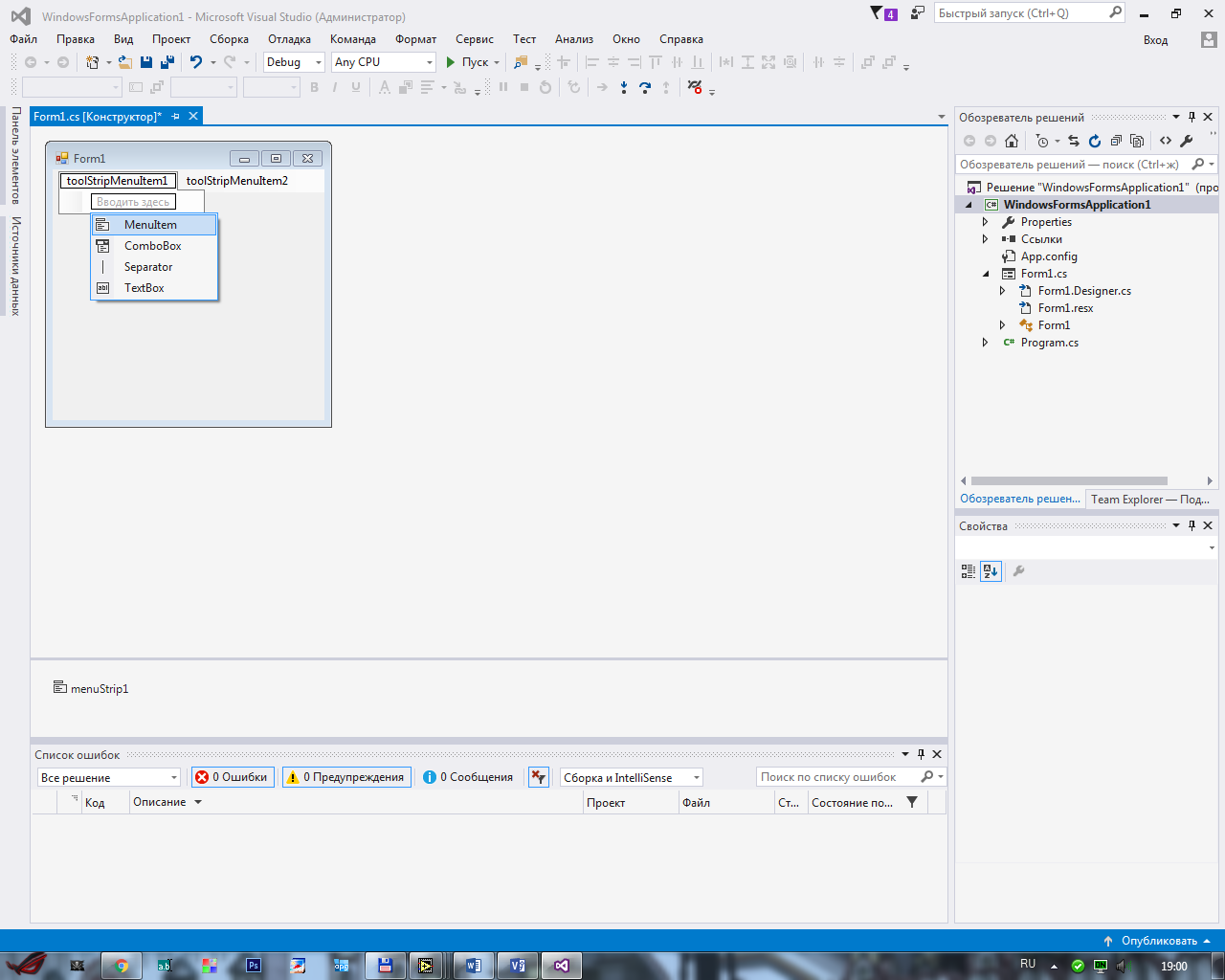


Рисунок 7 – В каждом пункте меню под каждый подпункт меню в конструкторе имеется задел, в котором можно выбрать варианты: подпункт меню, комбинированный список, разделитель, текстовое поле

**Элементы списка**

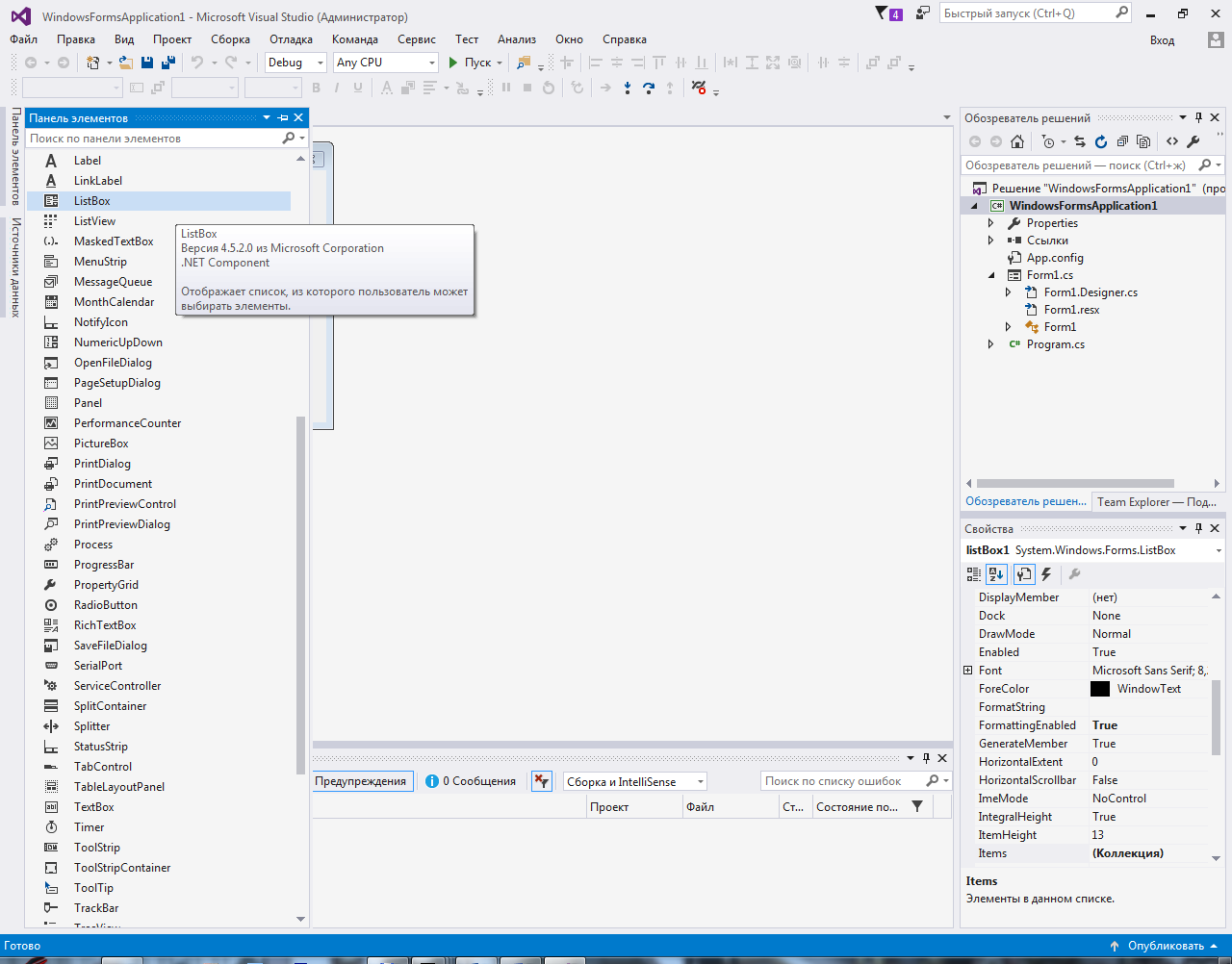


Рисунок 8 – Место расположения списка в панели элементов

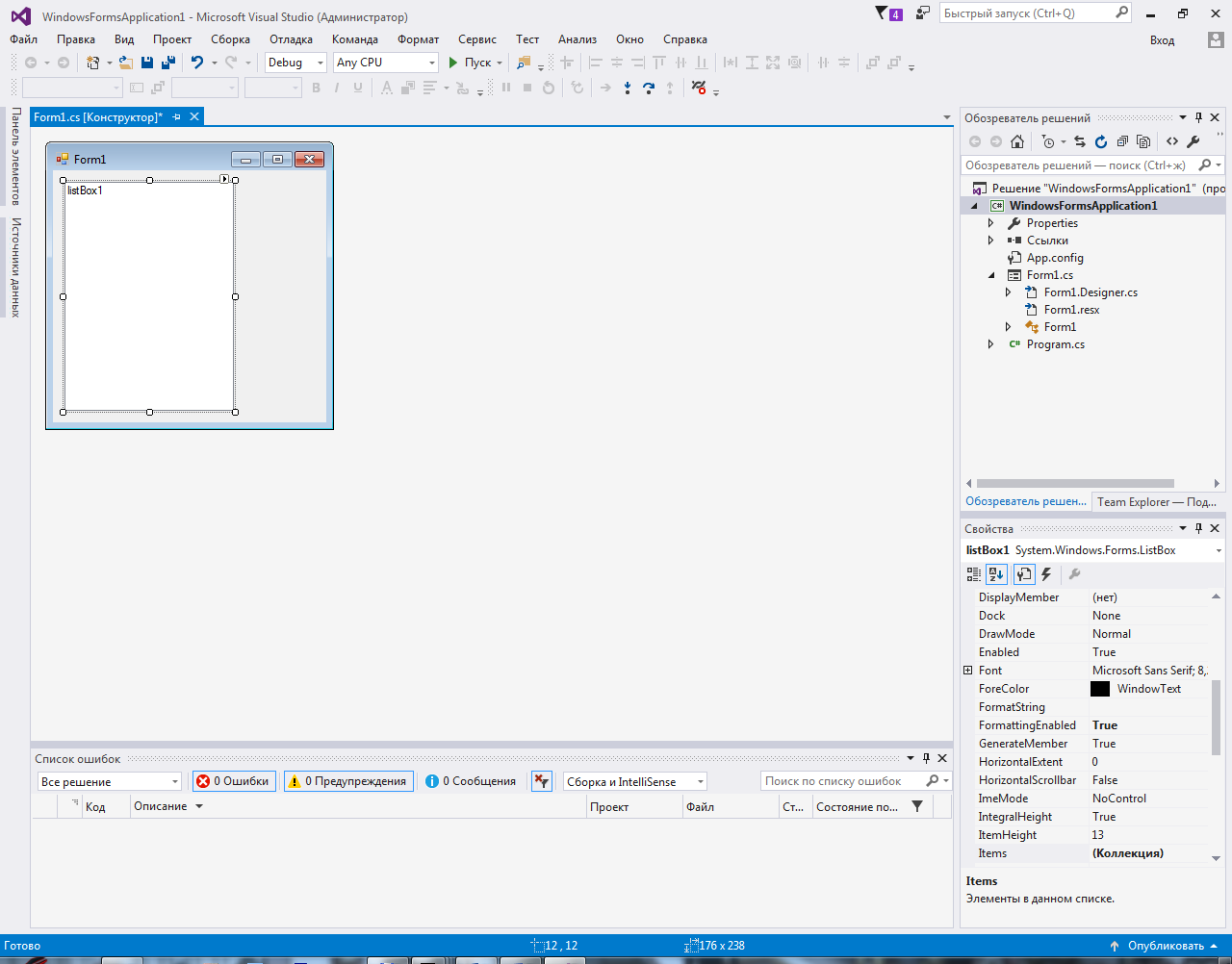


Рисунок 9 – Размещение списка на экранной форме

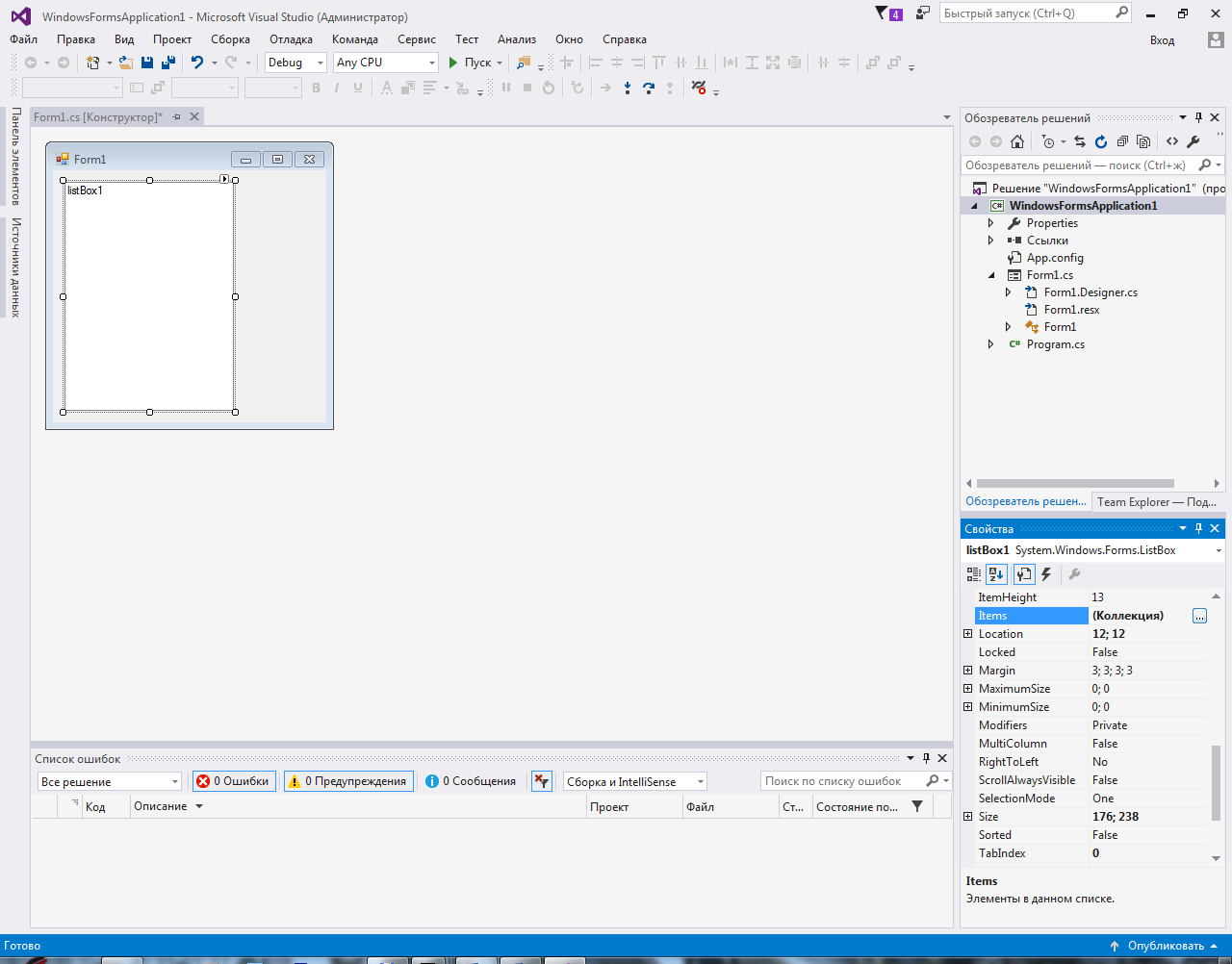


Рисунок 10 – Переход к редактированию содержимого списка (многоточие напротив пункта свойств *Items*)

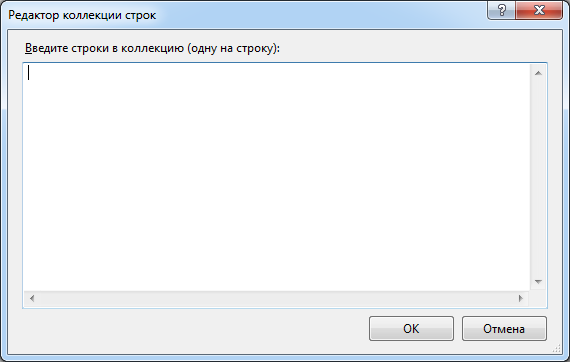


Рисунок 11 – В редакторе коллекции строк каждой физической строке соответствует пункт списка

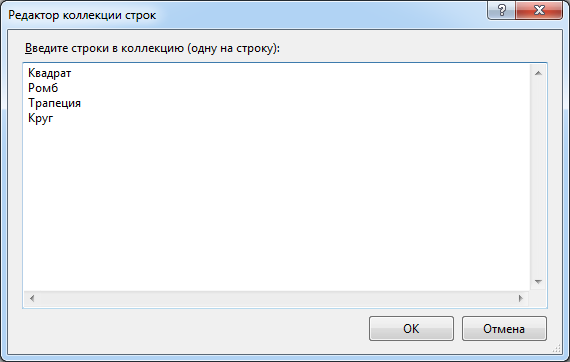


Рисунок 12 – Пример заполнения списка содержимым

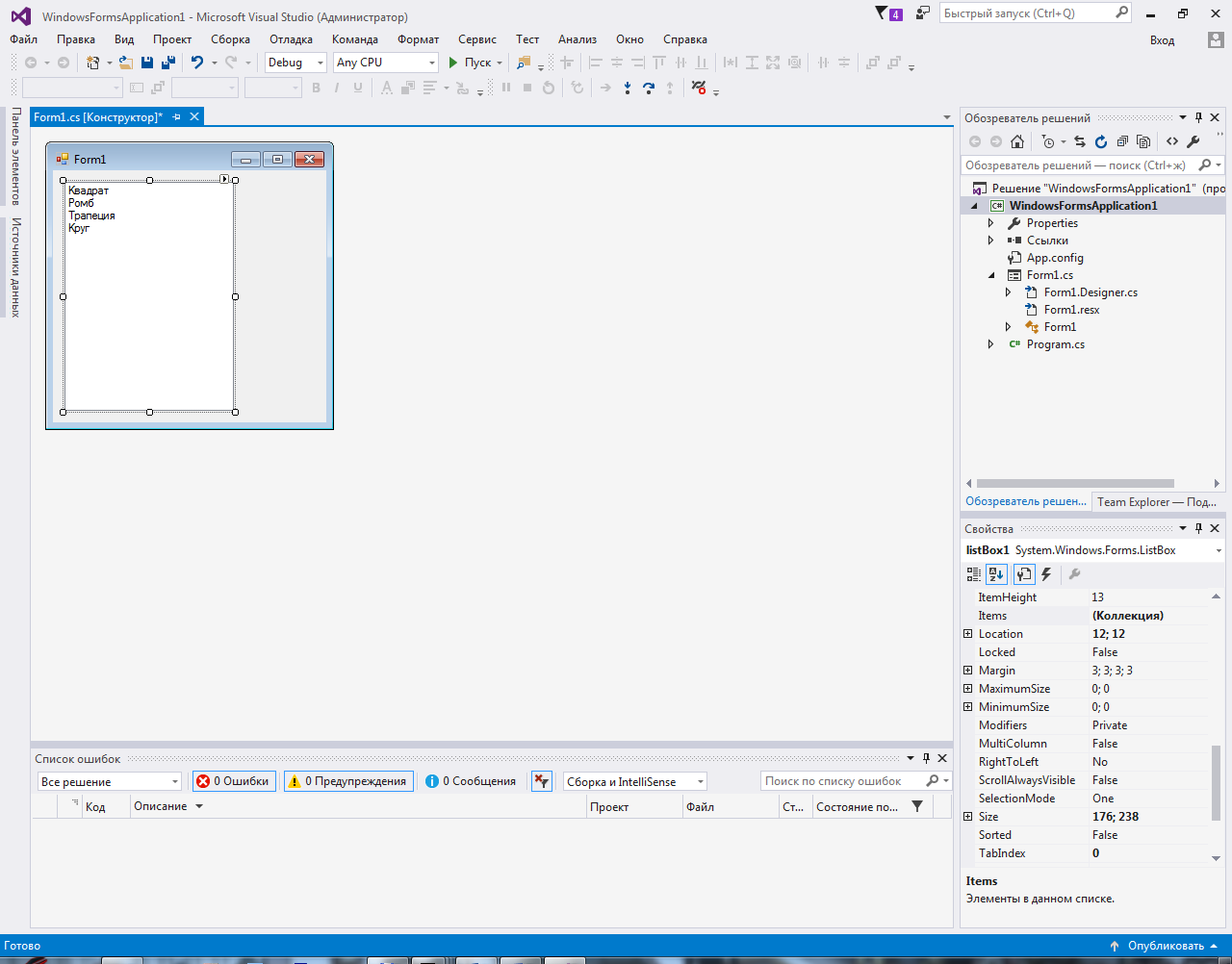


Рисунок 13 – Отображение введённого содержимого в списке

**Элементы комбинированного списка:**

Заполняется по аналогии с обыкновенным списком.

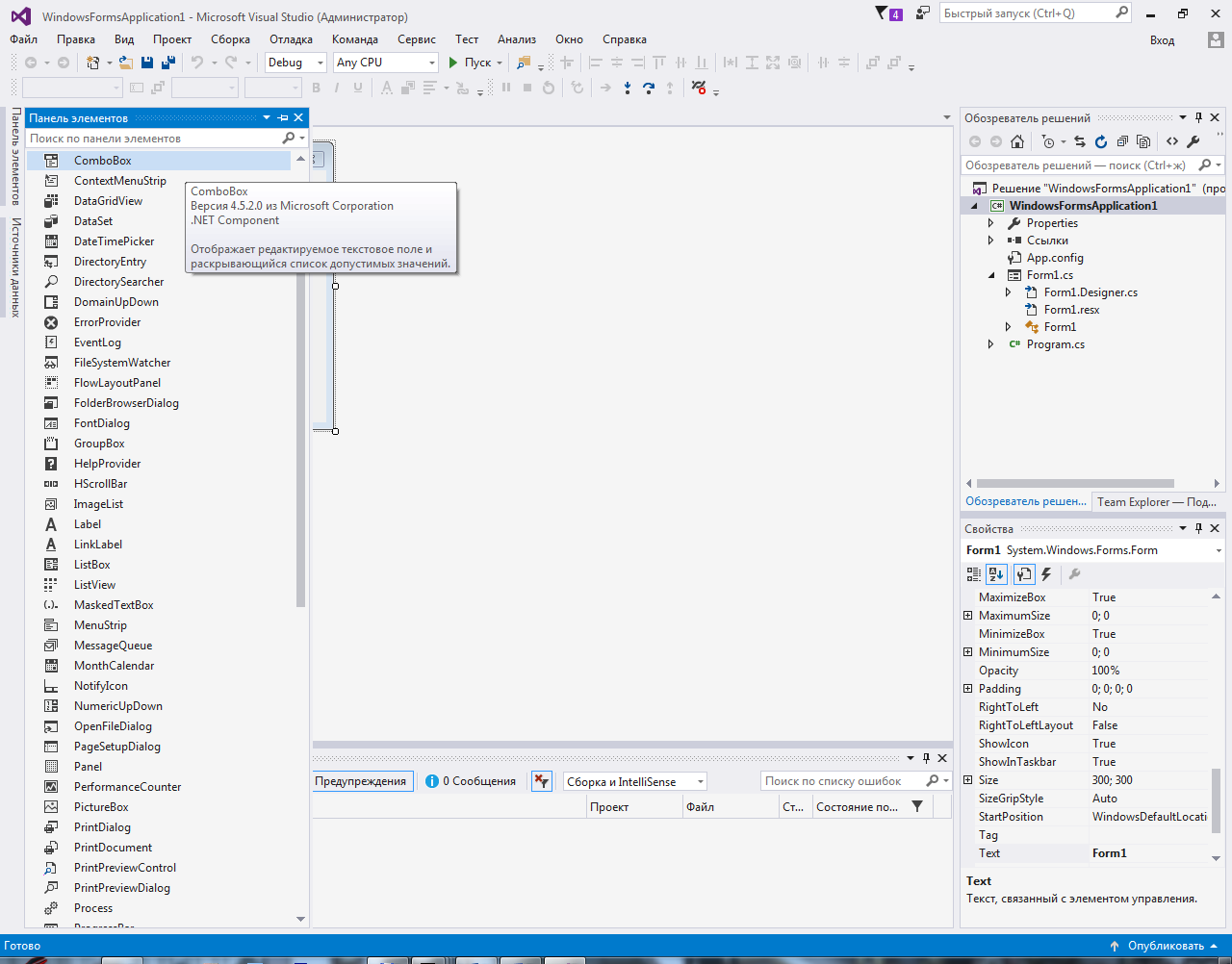


Рисунок 14 – Место расположения комбинированного списка в панели элементов

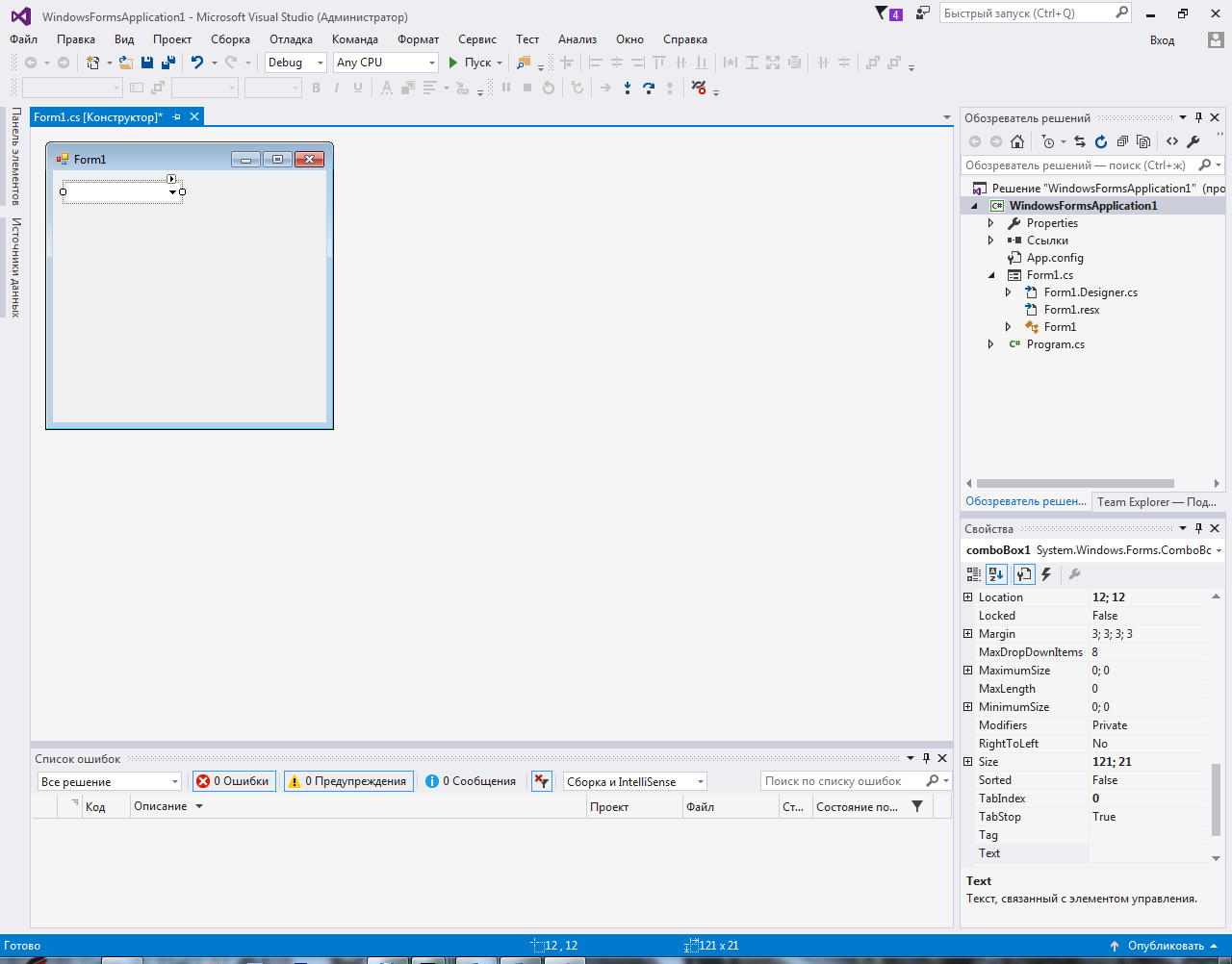


Рисунок 15 – Размещение комбинированного списка на экранной форме

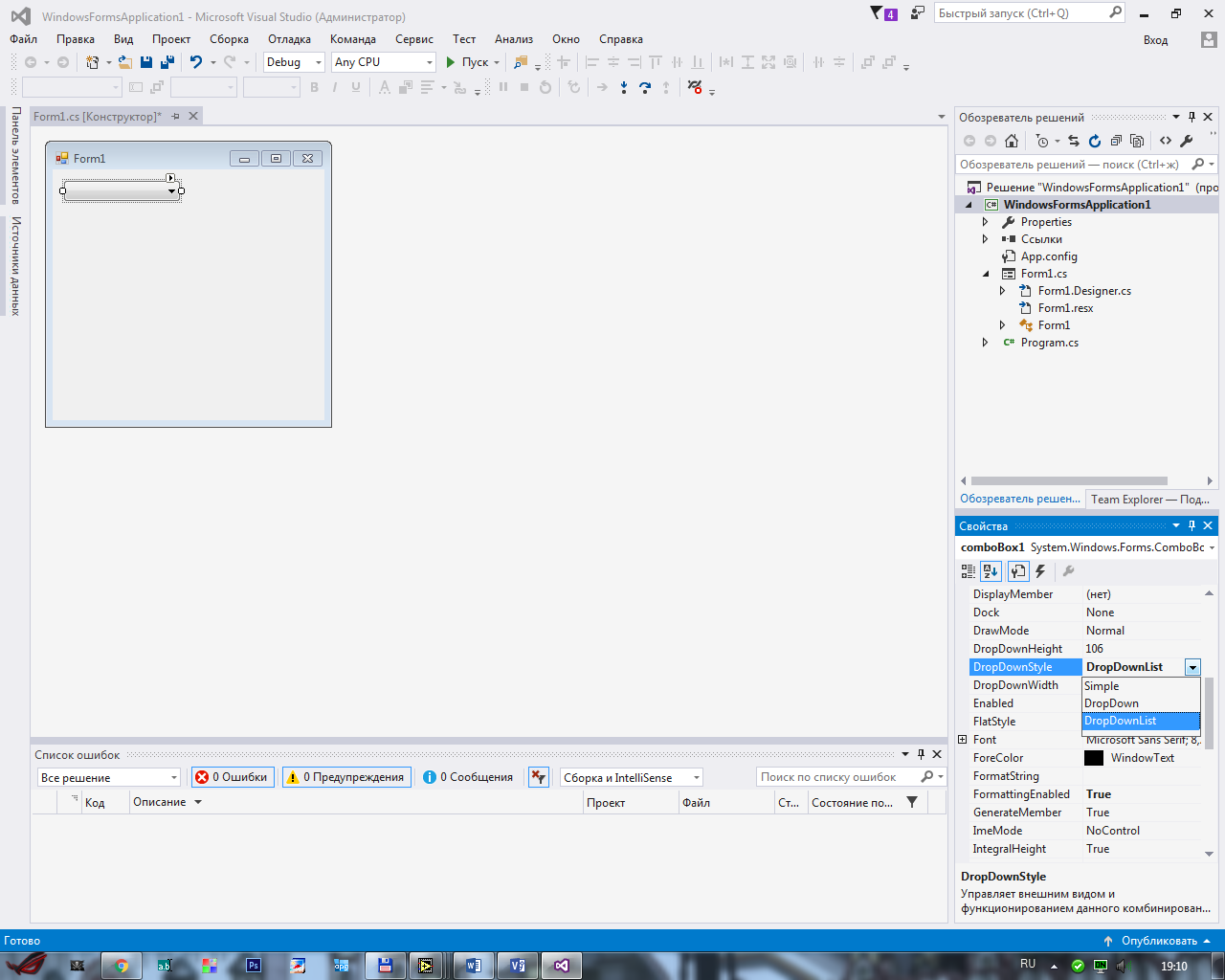


Рисунок 16 – Смена режима комбинированного списка на запрет ввода текста

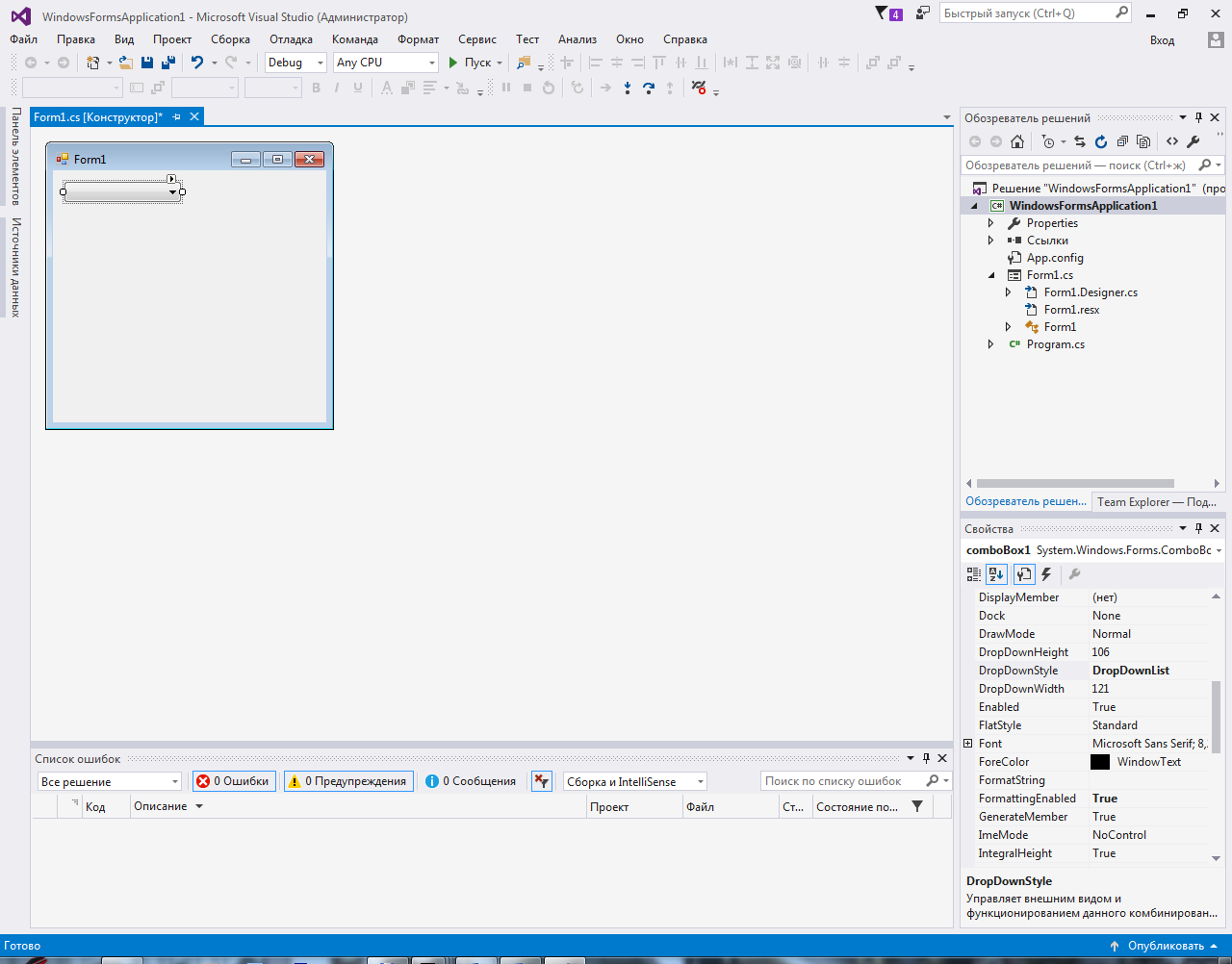


Рисунок 17 – Комбинированный список с запретом ввода текста, размещённый на экранной форме

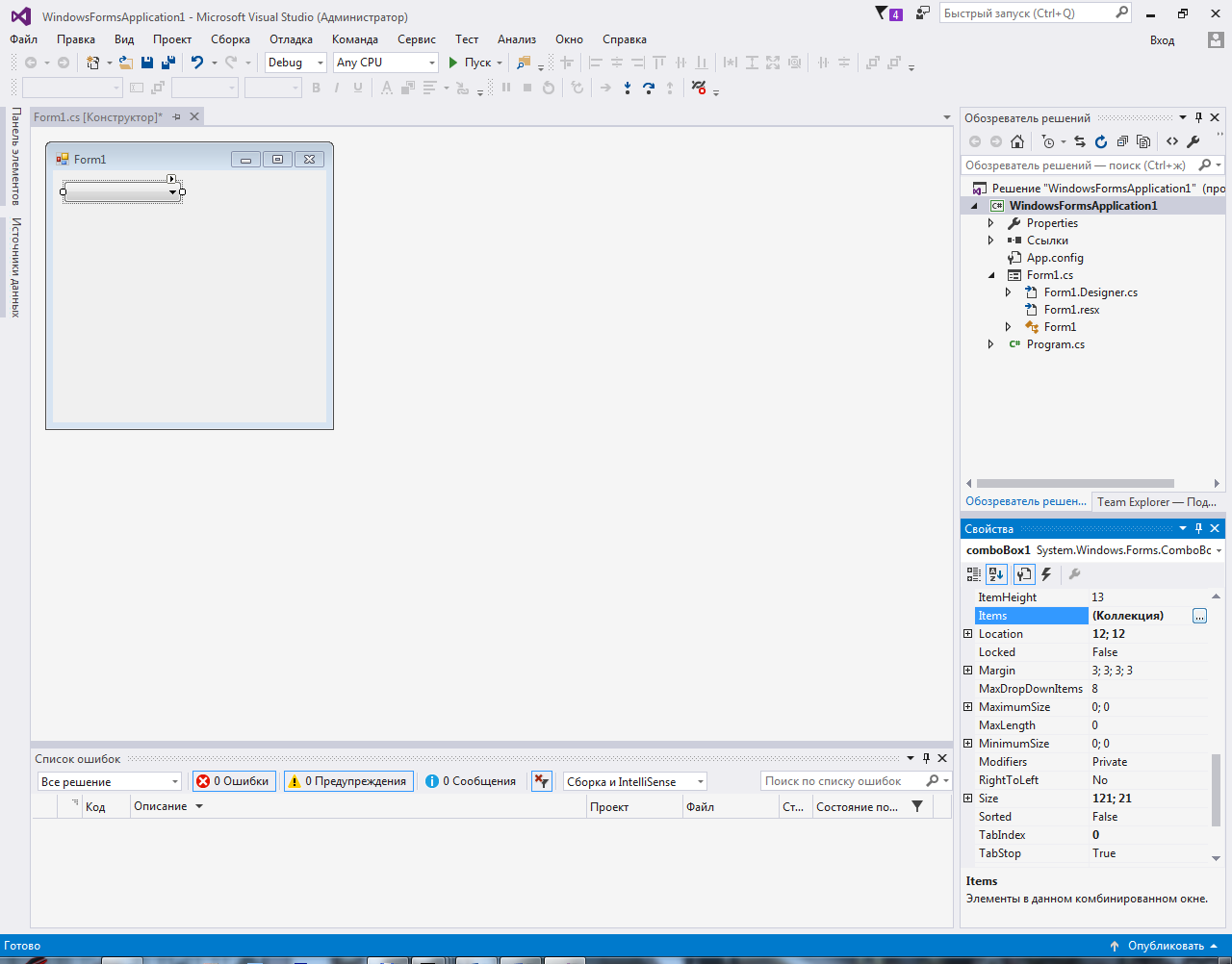


Рисунок 18 – Переход к редактированию содержимого списка (многоточие напротив пункта свойств *Items*)

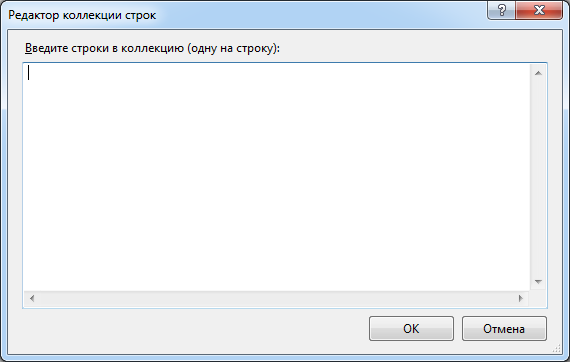


Рисунок 19 – В редакторе коллекции строк каждой физической строке соответствует пункт комбинированного списка

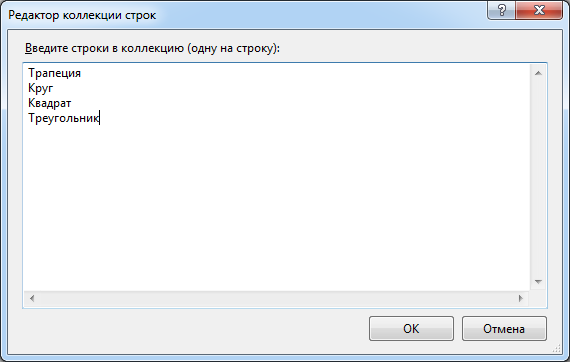


Рисунок 20 – Пример заполнения списка содержимым

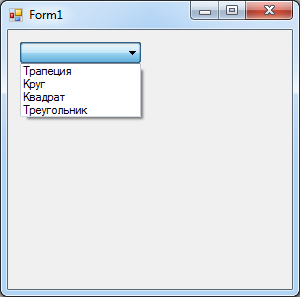


Рисунок 21 – Отображение содержимого комбинированного списка

**Требования к оформлению**:

1. Заголовок формы должен содержать надпись вида: «Задание №40 выполнил: [Фамилия И.О. автора]; Номер варианта: [Номер]; Дата выполнения: [дд/мм/гггг]».

2. Дата выполнения проставляется в момент, когда программа считается законченной и по ней можно готовить итоговый отчёт о выполнении работы.

3. Нечётные варианты отключают стандартный блок управления формой и создают авторские кнопки «Свернуть», «Развернуть», «Закрыть» внизу формы.

4. Чётные варианты заменяют стандартный курсор формы со «стрелки» на «руку».

5. Справочная информация должна быть вызвана в дочерней форме и считана в статический по размеру ярлык из текстового файла (нечётные варианты) или в статическое по размеру текстовое поле с выставленным запретом на редактирование текстовой информации (чётные варианты).

6. Нечётные варианты обеспечивают полупрозрачность дочерней формы.

7. Чётные варианты заменяют цвет подложки в соответствии с представленной ниже таблицей:

Таблица 1 – Таблица системных цветов подложки

|  |  |
| --- | --- |
| 2. | *ActiveCaption* |
| 4. | *AppWorkspace* |
| 6. | *ControlText* |
| 8. | *Desktop* |
| 10. | *GradientActiveCaption* |
| 12. | *Highlight* |
| 14. | *HotTrack* |
| 16. | *Info* |
| 18. | *MenuHighlight* |
| 20. | *Window* |

Подобрать читаемый цвет текста к предложенной по варианту схеме.

8. Исходное состояние всех элементов, расположенных на главной форме, должно быть настроено через перечень параметров этих элементов.

9. В качестве исходного состояния принимается заранее известная и заполненная элементами структура списков, все элементы формы за исключением главного меню находятся либо в недоступном состоянии (*.Enabled = false*), либо в невидимом состоянии (*.Visible = false*).

10. Кнопка «Начало работы с приложением» должна реализовывать активацию доступа пользователя к элементам или отображение элементов на форме для пользователя.

11. Все элементы программы должны носить значащие имена переменных, в которых отражено существо этих элементов, например, форма – *frmMain*, ярлык – *lblHelp*, комбинированный список – *cmbFigures* и т.д.

**Варианты размещения основных элементов:**

Таблица 2 – Таблица индексации фигур

|  |  |
| --- | --- |
| **№ вар.** | **Пункт порядок индексов фигур из списка** |
| **1** | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11. |
| **2** | 10, 9, 5, 1, 3, 11, 2, 6, 7, 8, 4. |
| **3** | 3, 2, 1, 11, 6, 9, 7, 8, 4, 10, 5. |
| **4** | 7, 3, 8, 4, 6, 2, 11, 9, 10, 5, 1. |
| **5** | 5, 4, 8, 2, 3, 1, 10, 9, 6, 11, 7. |
| **6** | 4, 3, 7, 5, 6, 2, 8, 9, 11, 10, 1. |
| **7** | 8, 6, 11, 2, 9, 10, 3, 4, 5, 1, 7. |
| **8** | 11, 7, 10, 6, 9, 3, 4, 8, 5, 2, 1. |
| **9** | 5, 3, 11, 4, 1, 9, 10, 6, 7, 8, 2. |
| **10** | 11, 8, 7, 5, 9, 10, 2, 4, 1, 6, 3. |
| **11** | 1, 7, 3, 9, 5, 11, 10, 4, 2, 6, 8. |
| **12** | 6, 7, 11, 8, 9, 3, 5, 4, 10, 2, 1. |
| **13** | 9, 1, 6, 7, 4, 11, 5, 10, 8, 3, 2. |
| **14** | 8, 2, 10, 7, 3, 11, 5, 9, 6, 1, 4. |
| **15** | 7, 5, 1, 4, 9, 8, 10, 3, 11, 6, 2. |
| **16** | 11, 2, 9, 4, 6, 5, 7, 8, 3, 10, 1. |
| **17** | 4, 3, 8, 7, 11, 2, 10, 9, 6, 1, 5. |
| **18** | 1, 7, 2, 6, 5, 9, 10, 3, 11, 4, 8. |
| **19** | 2, 1, 10, 11, 5, 9, 8, 6, 7, 3, 4. |
| **20** | 10, 9, 11, 8, 6, 5, 7, 4, 3, 2, 1. |

**Варианты индивидуального изображения:**

Таблица 3 – Сложные, комбинированные изображения

|  |  |  |
| --- | --- | --- |
| **№ вар.** | **Пункт списка** | **№ позиции** |
| **1** | Окружность единичного радиуса в декартовых координатах (оси со стрелками, текстом «0» на пересечении, но без засечек). | 7 |
| **2** | Крупная, центрированная надпись «Текст». | 10 |
| **3** | Правильный шестиугольник. | 3 |
| **4** | Трапеция с обводкой. | 1 |
| **5** | Прицел (две окружности, перекрестие и точка на перекрестии). | 8 |
| **6** | Дом (квадрат с маленьким квадратом с перекрестием внутри, треугольник равносторонний с маленьким кругом с перекрестием внутри). | 4 |
| **7** | Квадрат с обводкой и диагоналями. | 2 |
| **8** | Первая четверть декартовой системы координат (оси со стрелками, текстом «0» на пересечении, но без засечек). | 5 |
| **9** | Ромб с обводкой. | 11 |
| **10** | Крупная, центрированная надпись «Текст», заключённая в прямоугольник. | 9 |
| **11** | Олимпийские кольца. | 6 |
| **12** | Дом с трубой (квадрат с маленьким квадратом с перекрестием внутри, треугольник равносторонний с маленьким кругом с перекрестием внутри, прямоугольник). | 8 |
| **13** | Примитив человека (окружность, вертикальная линия, горизонтальная линия, две линии незавершённого равностороннего треугольника). | 10 |
| **14** | Воздушный шар на нитке с бликом. | 4 |
| **15** | Ромб с обводкой и диагоналями. | 1 |
| **16** | Воздушный змей на нитке. | 3 |
| **17** | Правильный десятиугольник. | 7 |
| **18** | Прямоугольный треугольник с медианой из прямого угла. | 11 |
| **19** | Прямоугольный треугольник с высотой из прямого угла. | 2 |
| **20** | Глаз (эллипс и окружности с обводками). | 9 |

**Варианты индивидуальных элементов управления:**

Таблица 4 – Элементы управления, необходимые для выбора изображаемых в *PictureBox* фигур

|  |  |
| --- | --- |
| **№ вар.** | **Элемент управления** |
| **1** | Пункты меню. |
| **2** | Список. |
| **3** | Комбинированный список. |
| **4** | Горячие клавиши. |
| **5** | Кнопки с иконками. |
| **6** | *CheckBox* и кнопка подтверждения. |
| **7** | Подпункты меню. |
| **8** | Опции *RadioButton*. |
| **9** | Поле для ввода текста и кнопка подтверждения. |
| **10** | Счётчик *NumericUpDown* и кнопка подтверждения |
| **11** | Список и кнопка подтверждения. |
| **12** | Комбинированный список и кнопка подтверждения. |
| **13** | Опции *RadioButton* и кнопка подтверждения. |
| **14** | Дублирование пунктов меню комбинированным списком. |
| **15** | Дублирование подпунктов меню списком. |
| **16** | Дублирование *CheckBox* пунктами меню. |
| **17** | Дублирование *RadioButton* подпунктами меню. |
| **18** | Счётчик *NumericUpDown*. |
| **19** | *CheckBox* |
| **20** | Дублирование *NumericUpDown* пунктами меню |