**Лабораторна робота № 7.**

***Тема:*** Створення **Windows Form**

**Мета роботи:**

Ознайомлення з функціями класів: деструктори, індексатори, операції класу, операції перетворення типів.

**Теоретичні відомості**

**Основи роботи з Visual Studio .NET**

Microsoft *Visual* Studio .*NET*- це інтегроване *середовище розробки* (Integrated *Development Environment* (*IDE*)) для створення, документування, завантаження та налагодження програм, написаних мовами .*NET*. Це потужний інструмент професійної розробки складних додатків, один із кращих. Набір його функцій надзвичайно великий. При завантажені *Visual* Studio .*NET*, те бачимо наступне вікно:

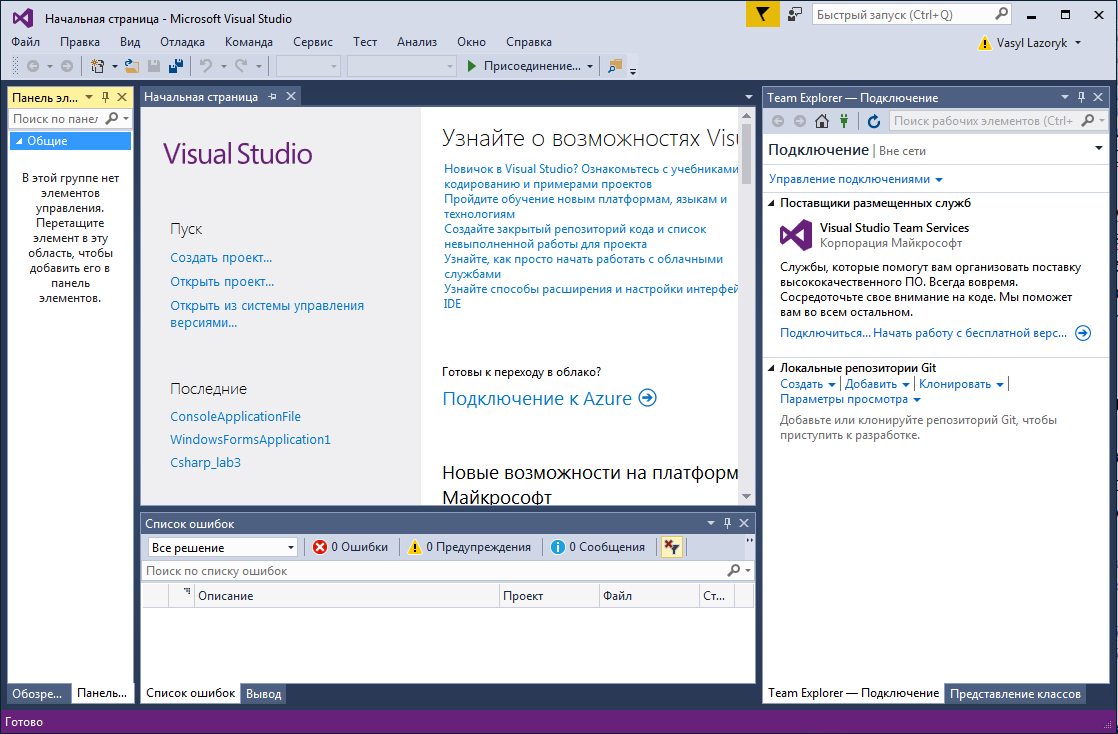
****

Рис.

Початкова сторінка (*Start Page*) містить наступні вкладки: Projects, *Online*Resources та My Profile. *За* замовчуванням вибраний *список* недавніх проектів. На цій же вкладці розташовані кнопки *New* *Projec ,* *Open* *Project*, тощо. На вкладці *Online* Resources відображаються *групи новин*, *заголовки* та посилання ресурсів розроблювачів. Ця *опція* доступна, коли *комп'ютер* підключений до Інтернету.

Главное вікно *Visual* Studio.*NET*, подібно іншим додаткам *Windows*, містить рядок *меню*, що включає в себе наступні категорії (коли ми перебуваємо на*Start Page*, частина категорій не видна – вона з'явиться пізніше, коли буде створений проект)

Строка меню Visual Studio .NET

**Рис.** Рядок меню Visual Studio .NET

У цих категоріях розташовані наступні команди:

* File - відкриття, створення, додавання, закривання, друк, тощо.
* Edit - стандартні команди виправлення: копіювання, вставка, вирізання , тощо.
* View - команди для приховання й відображення всіх вікон і панелей інструментів.
* Project - команди для роботи із проектом: додавання елементів, форм, посилань , тощо.
* Build - команди компіляції програми.
* Debug - команди для налагодження програми.
* Data - команди для роботи з даними.
* Format - команди форматування розташовуваних елементів (вирівнювання, інтервал , тощо).
* Tools - команди додаткових інструментів і настроювання Visual Studio .NET.
* Window - керування розташуванням вікон.
* Help - довідка.

Якщо після внесених змін і при наступному запуску *Visual* Studio .*NET* виявляється незвичний вид програми, можна налаштувати його заново, запустивши *Start Page* з *меню* Help/Show*Start Page*. Панелі, що ховаються, *розташовані* з боків вікна, закріпити їх на екрані, нажавши на значокЗначок , або зовсім забрати з екрана, а потім знову відобразити, використовуючи відповідний *пункт* *меню* View (або еквівалентне сполучення клавіш).

**Форми**

Форма — це екранний *об'єкт*, що забезпечує функціональність програми. Створюємо проект, який буде містити власні форми. Запускаємо *Visual* Studio .*NET*, вибираємо File/*New*/*Project* — з'являється *діалогове вікно* (Ctrl+*Shift*+N приводить до того ж результату), у якому вибираємо *Visual* *C*# *Project* й *Windows* *Application*

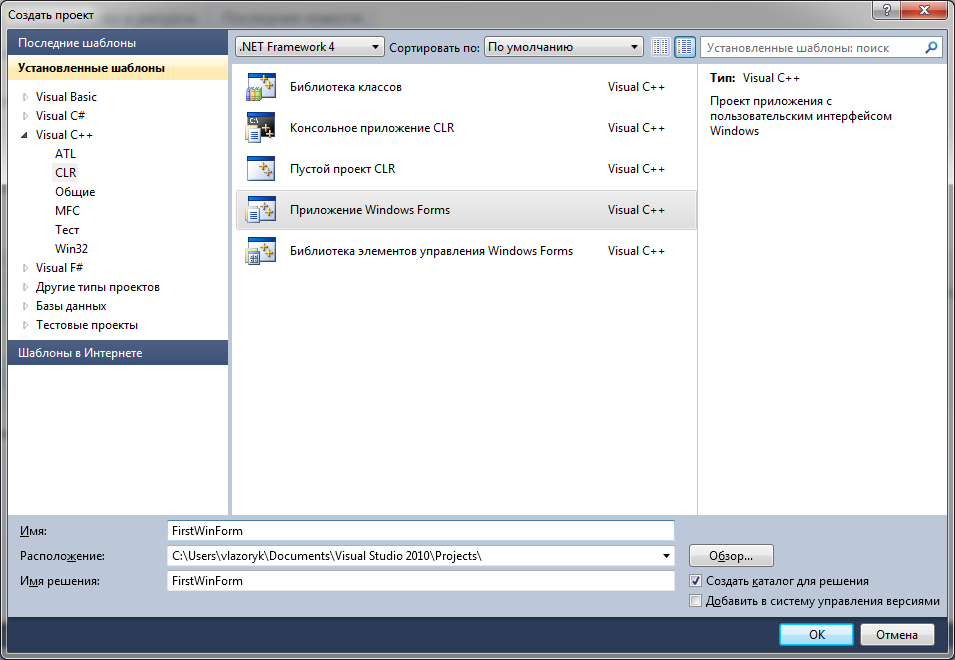
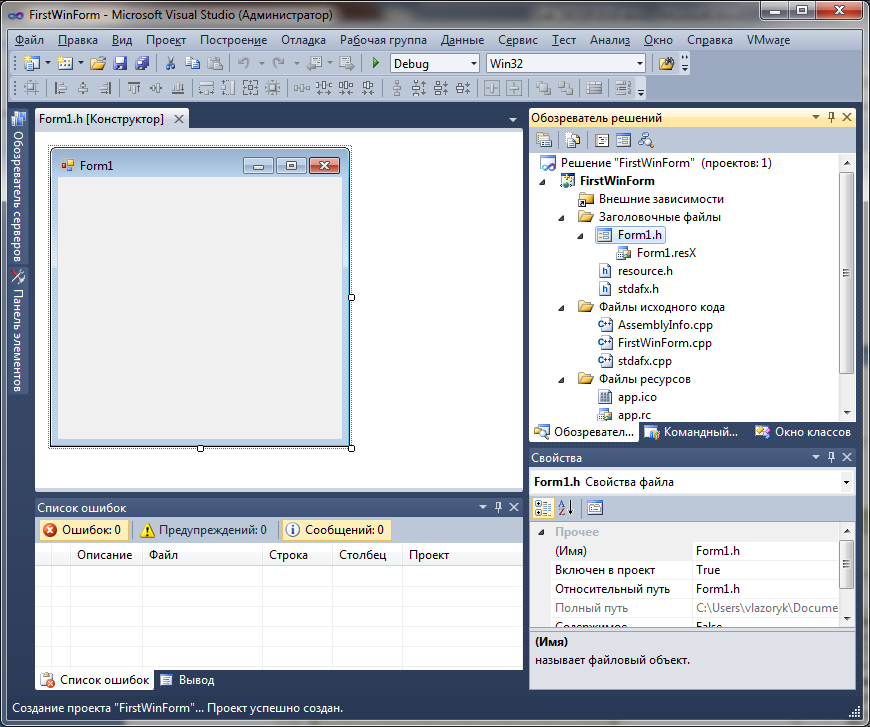


Рис. Створення нового проекту

У *поле* Name задаємо ім'я проекту — FirstForm і зберігаємо його в папку, обумовлену полем *Location*. Отриману папку можна згодом перемістити на інший *комп'ютер* і продовжити роботу — у ній будуть перебувати всі створювані нами файли цього проекту. На екрані з'явилася порожняя *Windows-форма*.



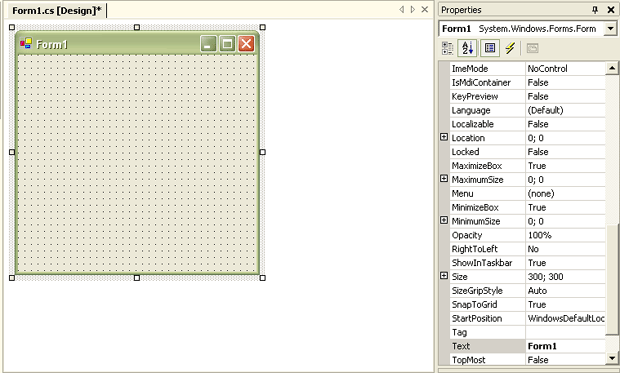
**Рис.** Головне вікно програми в режимі розробки **Solution Explorer**

Вікно *Solution Explorer* (провідник проекту, View —> *Solution Explorer*, або сполучення клавіш Ctrl+Alt +L) вміщує компоненти, що входять до складу проекту. При створенні нового проекту *Solution Explorer* містить компоненти, створені шаблоном. Можна одержувати короткий опис будь-якого методу, класу або властивості, просто клацнувши на ньому, — на інформаційній панелі негайно відобразиться коротка довідка. Для досвідченого розроблювача *Object* *Browser* — незамінний помічник у роботі, набагато більше зручний, чим довідка.

*Файл* *App*.ico містить зображення іконки, що на формі розташовано у верхньому лівому куті.

Вікно *Class* View — (огляд класів, View —>*Class*View, або сполучення клавіш Ctrl+*Shift*+*C*), дозволяє переміщатися в коді *по* обраному об'єкті; містить методи, класи, дані всього листинга проекту.

Вікно властивостей *Properties* — основний інструмент налаштування форми і її компонентів. Вміст цього вікна це *список* властивостей обраного в цей момент компонента або форми. Викликається це вікно декількома способами — у *меню* View вибираємо *пункт* Properties Window (або використаємо клавішу F4), на обраному об'єкті клацаємо правою кнопкою миші та у контекстному *меню* *пункт* Properties вибираємо *об'єкт* і натискаємо F4 або просто вибираємо *об'єкт* і переходимо у вікно Properties.

[](http://www.intuit.ru/EDI/14_11_15_12/1447519933-19070/tutorial/199/objects/1/files/01_11.png)

**Рис.**  Вікно властивостей Properties

Опис інтерфейсу самого вікна Properties.

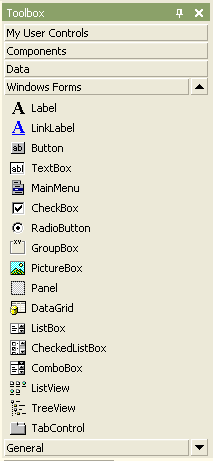
|  |  |  |
| --- | --- | --- |
| Таблиця 1.1. | | |
| **Елемент** | **Зображення** | **Опис** |
| Object name | Object name | У поле цього списку виводиться назва даного обраного об'єкта, що є екземпляром якого-небудь класу. Тут Form1 — назва форми за замовчуванням, що успадковується від классаSystem.Windows.Forms.Form |
| *Categorized* | Categorized | При натисканні на цю кнопку виробляється сортування властивостей обраного об'єкта по категоріях. Можна закривати категорію, зменшуючи число видимих елементів. Коли категорія схована, ви бачите знак (+), коли розкрита — (–) |
| Alphabetic | Alphabetic | Сортування властивостей і подій об'єкта за абеткою |
| Properties | Properties | При натисканні на цю кнопку відображається перерахування властивостей об'єкта |
| Events | Events | При натисканні на цю кнопку відображається перерахування подій об'єкта |
| Description Pane | Description Pane | Панель, на яку виводиться інформація про обрану властивість. У цьому випадку в списку властивостей форми було обране властивість Text |

Вікно Properties дозволяє визначати в першу *чергу* *дизайн* форми і її елементів керування. В таблиці 1.2 приводиться опис деяких властивостей форми, звичайно обумовлених у режимі дизайну. При виборі значення властивості, відмінного від прийнятого *за* замовчуванням, воно виділяється жирним шрифтом, що полегшує надалі *визначення* змін.

|  |  |  |
| --- | --- | --- |
| Таблиця 1.2. Деякі властивості форми | | |
| **Властивість** | **Опис** | **Значення за замовчуванням** |
| Name | Назва форми в проекті. Це не заголовок форми, що ви бачите при запуску форми, а назва форми усередині проекту, що ви будете використати в коді | Form1, Form2 і т.д. |
| AcceptButton | Установлюється значення кнопки, що буде спрацьовувати при натисканні клавіші Enter. Для того щоб ця властивість була активним, необхідна наявність принаймні однієї кнопки, розташованої на формі | None |
| BackColor | Кольори форми. Для швидкого перегляду різних варіантів просто клацайте прямо на назві "BackColor" | Control |
| BackgroundImage | Зображення на заднім тлі | None |
| CancelButton | Установлюється значення кнопки, що буде спрацьовувати при натисканні клавіші Esc. Для того щоб ця властивість була активним, необхідна наявність принаймні однієї кнопки, розташованої на формі | None |
| ControlBox | Установлюється наявність або відсутність трьох стандартних кнопок у верхньому правому куті форми: "Згорнути", "Розгорнути" й "Закрити" |  |
| Cursor | Визначається вид курсору при його положенні на формі | Default |
| DrawGrid | Установлюється наявність або відсутність сітки із крапок, що допомагає форматувати елементи керування. У кожному разі сітка видна тільки на стадії створення додатка | True |
| Font | Форматування шрифту, використовуваного для відображення тексту на формі в елементах керування | Microsoft Sans Serif ; 8,25pt |
| FormBorderStyle | Визначення виду границь форми. Можливі варіанти:   * None — форма без границь і рядка заголовка; * FixedSingle — тонкі границі без можливості зміни розміру користувачем; * Fixed3D — границі без можливості зміни розміру із тривимірним ефектом; * FixedDialog — границі без можливості зміни, без іконки додатка; * Sizable — звичайні границі: користувач може змінювати розмір границь; * FixedToolWindow — фіксовані границі, є тільки кнопка закриття форми. Такий вид мають панелі інструментів у додатках; * SizableToolWindow — границі з можливістю зміни розмірів, є тільки кнопка закриття форми | Sizable |
| Icon | Зображення іконки, розташовуваної в заголовку форми. Підтримуються формати .ico | Icon |
| MaximizeBox | Визначається активність стандартної кнопки "Розгорнути" у верхньому правому куті форми | True |
| MaximumSize | Максимальний розмір ширини й висоти форми, що задає в пикселях. Форма буде приймати зазначений розмір при натисканні на стандартну кнопку "Розгорнути" | 0;0 (У весь екран) |
| MinimizeBox | Визначається активність стандартної кнопки "Згорнути" у верхньому правому куті форми | True |
| MinimumSize | Мінімальний розмір ширини й висоти форми, що задає в пикселях. Форма буде приймати зазначений розмір при зміні її границь користувачем (якщо властивість FormBorderStyle має значення за замовчуванням Sizable) | 0;0 |
| Size | Ширина й висота форми | 300; 300 |
| StartPosition | Визначення розташування форми при запуску додатка. Можливі наступні значення:   * Manual — форма з'являється у верхньому лівому куті екрана; * CenterScreen — у центрі екрана; * WindowsDefaultLocation — розташування форми за замовчуванням. Якщо користувач змінив розміри форми, то при наступному її запуску вона буде мати той же самий вид і розташування; * WindowsDefaultBounds — границі форми приймають фіксований розмір; * CenterParent — у центрі батьківської форми | WindowsDefaultLocation |
| Text | Заголовок форми. На відміну від властивості Name, саме ця назва форми, що не використається в коді | Form1, Form 2 і т.д. |
| *WindowState* | Визначення положення форми при запуску. Можливі наступні значення:   * Normal — форма запускається з розмірами, зазначеними у властивості Size; * Minimized — форма запускається з мінімальними розмірами, зазначеними у властивості MinimumSize; * Maximized — форма розвертається на весь екран | Normal |

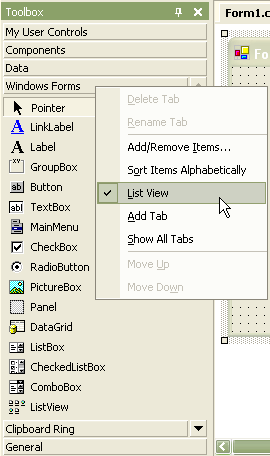
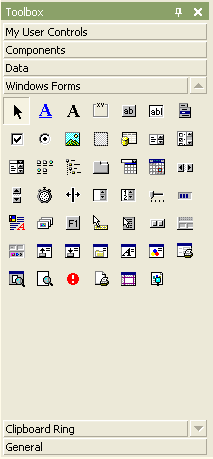
Кнопка вікна властивостей Events (Події)Кнопка перемикає вікно Properties у режим керування оброблювачами різних подій (наприклад, миші, клавіатури) і одночасно виводить *список* всіх подій компонента. *Подвійний щиглик* миші в *поле* значення події генерує оброблювач для нього й перемикає в режим коду.

Вікно *Toolbox* (*панель інструментів*, View —> *Toolbox*, або сполучення клавіш Ctrl+Alt+X) містить компоненти *Windows-форм*, називані також елементами керування, які розміщаються на формі. Воно складається з декількох закладок: My *User* Controls, Components, *Data*, *Windows* Forms й *General* ([рис. 1.11](http://www.intuit.ru/studies/courses/106/106/lecture/28450?page=3#image.1.11)):



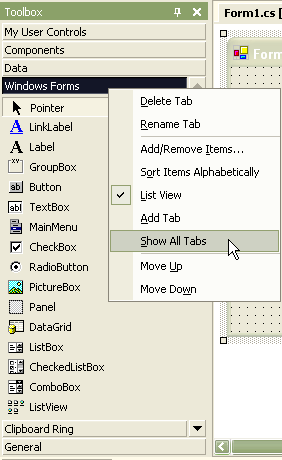
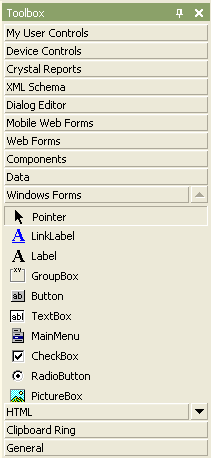
**Рис.** Вікно Toolbox

Найбільше часто вживаною закладкою є *Windows* Forms. Для розміщення потрібного елемента керування досить просто клацнути на ньому у вікні *Toolbox* або, схопивши, перетягнути його на форму. Перемикання виду значків дозволяє розмістити їх без смуги прокручування

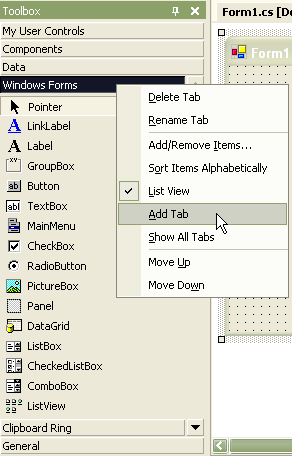
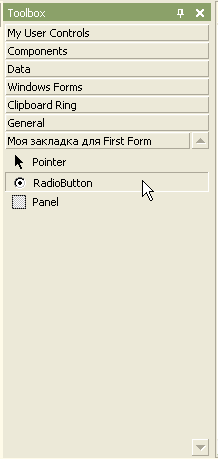
**Рис.** Подання елементів у вигляді списку значків

У вікні *Toolbox* доступне *відображення* всіх закладок.

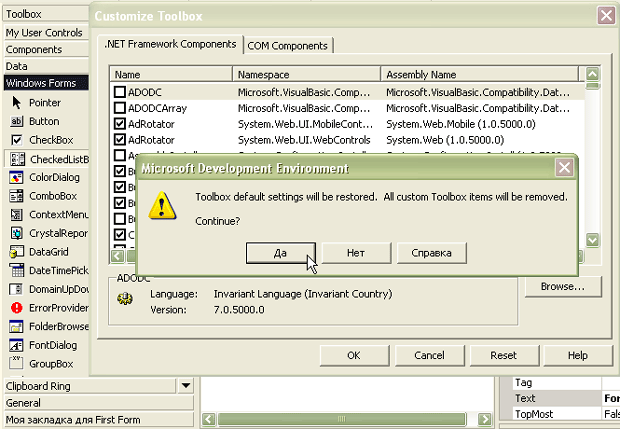
 . 

**Рис.** Установлення галочки "Показати всі закладки" та повний список закладок

Закладка My *User* Controls дозволяє зберігати власні списки елементів керування — якщо ви найбільше часто використаєте лише трохи з них, мабуть, має сенс перетягнути на цю закладку потрібні елементи. Або створити свою власну закладку

**Рис. 1.16.** Додавання закладки

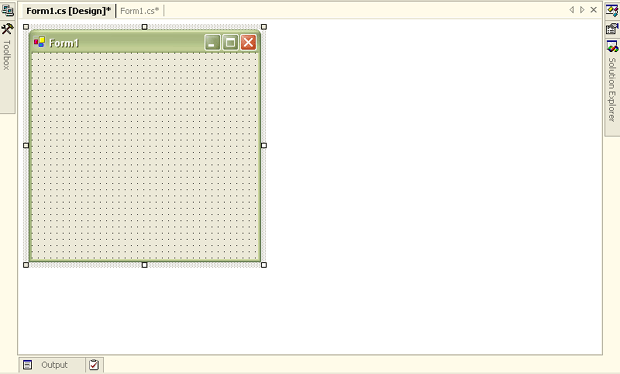
[](http://www.intuit.ru/EDI/14_11_15_12/1447519933-19070/tutorial/199/objects/1/files/01_26.png)

**Рис. 1.18.** Відновлення значень за замовчуванням

Створені в такий спосіб закладки можна перейменувати або видалити, вибравши в контекстному *меню* пункти *Rename* *Tab* й *Delete* *Tab* відповідно.

Якщо в результаті всіх експериментів ви виявите, що поточний вид вікна *Toolbox* сильно відрізняється від первісного, для відновлення значень *за* замовчуванням виберіть у контекстном*меню* будь-якої закладки *пункт* Add/Remove Items…... У вікні, що з'явилося, натисніть на кнопку Reset. З'являється *вікно попередження* — "Настроювання *Toolbox* будуть відновлені. Всі користувальницькі закладки будуть вилучені. Продовжувати?" Погодившись із попередженням, ви побачите вид *Toolbox* *за* замовчуванням.

*Продуктивність* розробки додатка багато в чому залежить від зручності настроювання користувальницького середовища. Одним з ергономічних варіантів уважається загальне розташування вікон, що ховаються, що не захаращує головну частину проекту[рис. 1.19](http://www.intuit.ru/studies/courses/106/106/lecture/28450?page=3#image.1.19)).

[](http://www.intuit.ru/EDI/14_11_15_12/1447519933-19070/tutorial/199/objects/1/files/01_27.png)

**Рис. .** Ергономічне розташування вікон, максимально звільняючу робочу область

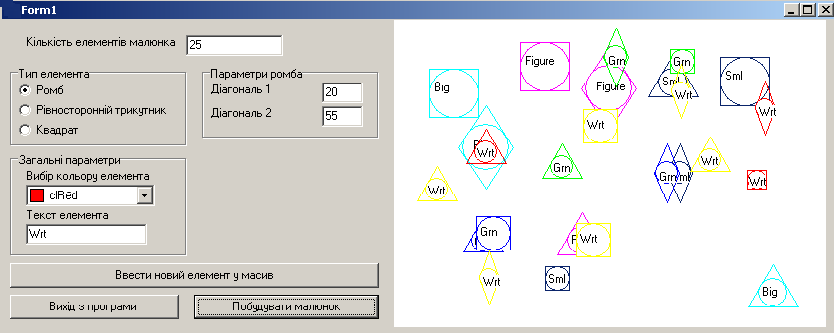
**Режими дизайну й коду**

При створенні нового проекту завантажується режим дизайну — форма являє собою основу для розташування елементів керування. Для роботи із програмою варто перейти в режим коду. Це можна зробити декільком способами: клацнути правою кнопкою миші в будь-якій частині форми й у *меню, що* з'явився, вибрати View Code, у *вікні Solution* Explorer зробити те ж саме на компоненті Form 1.cs або просто двічі клацнути на формі — при цьому згенерується метод Form1\_Load. Після хоча б однократного переходу в режим коду в цьому проекті з'явиться вкладка Form1.cs\* натискаючи на яку, теж можна переходити в режим коду. Для переходу в режим коду також можна використати клавішу F7, а для повернення в режим дизайну — сполучення *Shift*+F7.

***Завдання до лабораторної роботи:***

**Порядок виконання роботи:**

1. Створити проект C++/clr **Windows Form**.
2. Розробити програму для вирішення задачі.
3. Підготувати звіт у твердій копії та в електронному виді.



1. Створити форму для виконання задачі. Написати програму з використання **Windows Form** виведення малюнка, згідно варіанту. В програмі розробити форму для ведення даних про об’єкти малюнка (тип об’єкта, розміри, колір тощо) та для виведення малюнка в об’єкт Image. Малюнок будується як набір елементів(типи елементів задаються згідно варіанту), координати розташування елемента в об’єкт PictureBox задавати за допомогою датчика випадкових чисел в межах полотна об’єкт PictureBox. Кожний елемент є об’єктом одного з похідних класів. Набір – задається, як масив на абстрактний базовий клас. В базовому класі передбачити віртуальні функцію малювання, функцію переміщення та інші функції. Базовий клас фігура похідні класи :
   1. Квадрат, прямокутник, еліпс та ромб.
   2. Дуга, сектор, еліпс та прямокутник із заокругленими кутами .
   3. Квадрат, прямокутник, трикутник та коло.
   4. Коло, квадрат, правильний трикутник та зірку.
   5. Круг, сектор, зафарбований прямокутник та зірку.
   6. Коло, квадрат та прямокутник з текстом в середині.
   7. Квадрат, правильний трикутник та коло з текстом в середині.
   8. П’ятикутник, прямокутник, трикутник та ромб.
   9. Зафарбований еліпс, трикутник , прямокутник та зірку.
   10. Шестикутник, ромб, трикутник та дугу.
   11. Прямокутник із заокругленими кутами, еліпс, дуга та квадрат.
   12. Прямокутник із заокругленими кутами, ромб, коло та дугу.
   13. П’ятикутник, ромб та правильний трикутник з текстом в середині.
   14. Правильні п’яти – , шести – та восьмикутники.
   15. Шестикутник, ромб та коло з текстом в середині.
   16. Квадрат, трапеція та ромб вписані в коло.
   17. Прямокутник, правильний п’ятикутник та трикутник вписані в коло.

**Контрольні вопросы:**

1. Як створити нове Windows додаток в середовищі .NET?

2. Як додати новий компонент на форму?

3. Які основні властивості і події має форма?

4. Які основні властивості і події мають елементи управління?

5. Як можна визначити поточний стан курсору мишки на формі?