

## Вступ

**Тема:** Знайомство з інтегрованим середовищем програмування C++ BUILDER

**Мета:** оволодіти практичними навичками роботи в інтегрованому середовищі програмування системи Borland C++

### ТЕОРЕТИЧНІ ВІДОМОСТІ

#### 1. Вікно середовища C++ Builder

C++ Builder - це технологія візуального програмування, де автоматизовано її трудомістку частину - створення інтерфейсних програм з діалоговими вікнами. C++ Builder призначено для написання програм мовою програмування C++ і поєднує VCL та середовище програмування (IDE - Integrated Development Environment), написані на Delphi з компілятором C++.

Вікно середовища C++ Builder при завантаженні складається з таких елементів (рис.):

- вікно форми;
- вікно коду програми;
- головне меню;
- “гарячі” кнопки інструментальних панелей;
- палітра компонентів;
- вікно інспектора об’єктів.

*Вікно форми* займає найбільше місце і є прямокутним сірим “контейнером” (рис.), на який при проектуванні форми розміщують компоненти (кнопки, надписи, панелі, вікна редакторів тощо). Форма сама є компонентом з назвою Form. Без додаткових вказівок заголовок компонента (властивість Caption) збігається з його назвою (властивість Name), до якої додається порядковий номер, розпочинаючи з 1. Але заголовок можна змінити за допомогою властивості Caption. Розміри форми також можна змінювати чи то за допомогою вікна Object Inspector, чи просто “зачепивши” мишкою за лінію межі форми (цього моменту курсор набуває вигляду двонапрямленої

стрілки).

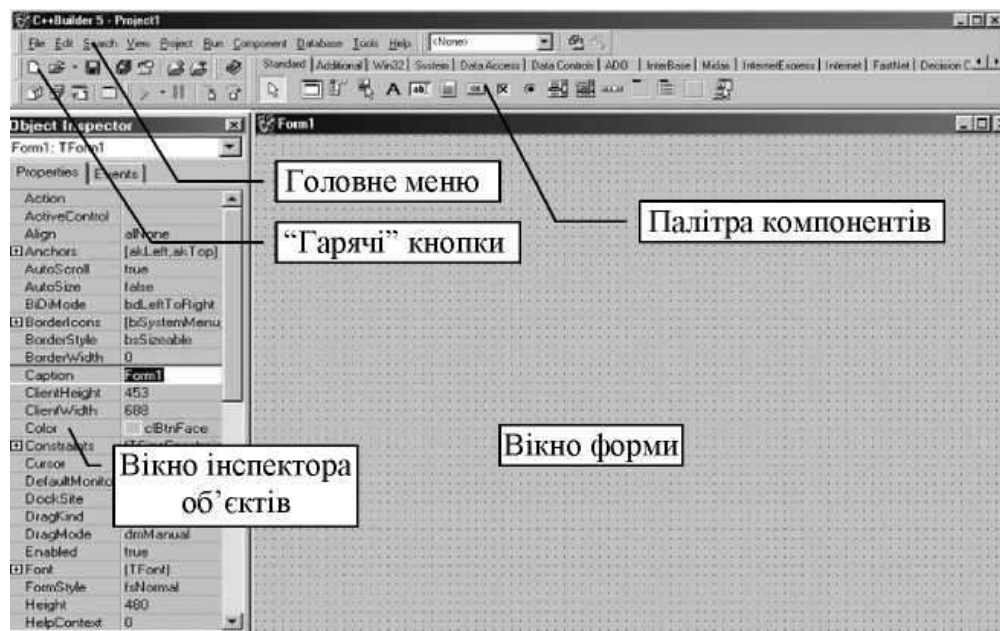


Рис. Вигляд головного вікна середовища C++ Builder

Для розміщених на формі компонентів можна викликати контекстну довідку, для чого слід виокремити потрібний компонент і натиснути клавішу <F1>. Якщо клацнути на самій формі і натиснути клавішу <F1>, відкриється довідка по класу форми TForm.

Вікно редактора коду програми Unit1.cpp (рис). Саме в цьому вікні прописують текст програми. Зазвичай вікно редактора коду перекривається вікном форми. Перемикання поміж вікнами форми й редактора можна здійснювати клавішею <F12> чи то натисненням миші на видимій частині обраного вікна.

З самого початку редактор коду містить мінімальний набір інструкцій, що забезпечує нормальне функціонування порожньої форми у якості Windows-вікна. При створенні проекту програміст вносить до нього потрібний програмний код. Більш докладно структуру програмного модуля розглянуто пізніше.

У вікно редактора коду, як і в інші вікна C++ Builder, вбудовано контекстну довідку. Щоб отримати довідку стосовно певного слова (терміну)

коду, достатньо встановити курсор на це слово і натиснути клавішу <F1>.

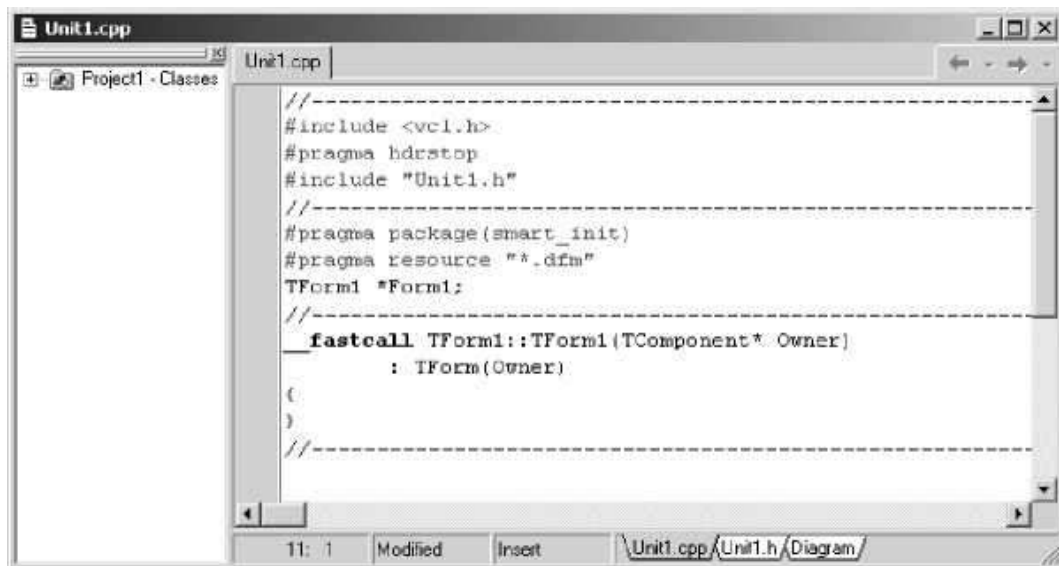


Рис. Вигляд вікна редактора коду

## 2. Головне меню

Головне меню міститься у верхній частині основного вікна (рис.). Воно надає можливість доступу до всіх інструментальних засобів створення, модифікації та керування проектом. Меню містить багато підменю (File, Edit, Search тощо). Слід звернути увагу лише на основні, найбільш використовувані опції:

Підменю File (Файл) містить команди для виконання операцій з проектами, модулями та файлами. Його основними опціями є:

- New Application - створити новий проект;
- New Form\* - створити й долучити до проекту нову форму;
- Save\* - зберегти відкритий активний модуль (unit);
- Save As- зберегти активний модуль (unit) під іншим ім'ям;
- Save Project As - зберегти проект під іншим ім'ям;
- Save All\* - зберегти файл проекту та всі його модулі;
- Close - закрити поточний модуль і відповідну форму;
- Close All - закрити усі відкриті вікна проекту.

*Примітка.* Опції, позначені символом \*, також можна реалізовувати за

допомогою “гарячих” кнопок, розглянутих в табл.

Підменю Edit (Правка) містить команди редагування, роботи з областю обміну даними, відміни дій і керування відображенням компонентів. Його основними опціями є:

Undo, Redo - відмінити чи відновити останню зміну (еквівалентні дії викликаються комбінаціями клавіш: <Ctrl>+<Z> і <Shift>+<Ctrl>+<Z>);

Cut, Copy - вирізати чи лише скопіювати виокремлені компоненти чи фрагмент тексту з модуля до буфера (<Ctrl>+<X> і <Ctrl>+<C>);

Paste- вставити з буфера дані (<Ctrl>+<V>), які були до цього скопійовані чи вирізані в буфер;

Delete - вилучити виокремлені компоненти форми чи тексту (клавіша <Delete>);

Select All - виокремити всі компоненти форми чи увесь текст по точного модуля;

Aline\* - викликати вікно вирівнювання виокремлених компонентів форми;

BringToFront\* - перемістити виокремлений компонент на передній план;

SendToBack\* - перемістити виокремлений компонент на задній план;

Scale\* - викликати вікно для пропорційного змінення масштабу виокремлених компонентів.

*Примітка.* Опції, позначені символом \*, також можна реалізовувати за допомогою допоміжного меню, котре відкривається за натиснення правої кнопки миші на виокремленому компоненті.

Для решти підменю головного меню зазначимо лише область їхньої дії, оскільки при початковому вивченні C++ Builder та створенні нескладних проектів можна користуватися значеннями опцій за замовчуванням чи використовувати “гарячі” кнопки чи то комбінації клавіш клавіатури.

Підменю Search (Пошук) містить команди для пошуку (<Ctrl>+<F>) і заміни (<Ctrl>+<R>) фрагментів тексту, а також покрокового пошуку рядків з помилками компілювання. Окрім того, можна здійснювати пошук за номером

сторінки чи за певною послідовністю символів.

Підменю View (Вид) містить команди для виклику менеджера проектів, інспектора об'єктів (<F11>), браузера об'єктів, вікон налагодження та інших інформаційних утиліт.

Підменю Project (Проект) містить команди компіляції (<Ctrl>+<F9>) та конструювання проектів, а також встановлення опцій поточного проекту.

Підменю Run (Пуск) містить команди для запуску проекту на виконання (<F9>), засобів контролю покрокового виконання і налагоджування програм.

Підменю Component (Компоненти) містить команди для створення та інсталяції нових компонентів на базі вже існуючих, а також для налаштування конфігурації палітри компонентів.

Підменю DataBase (Бази даних) містить команди виклику інструментальних засобів для роботи з базами даних.

Підменю Tools (Інструменти) містить команди для налаштування середовища та виклику додаткових утиліт, у тому числі вікно редактора графічних зображень, інструменти обслуговування бази даних і вікно для редагування під-меню Tools.

Підменю Help (Допомога) містить команди для виклику різних розділів довідкової системи. Зокрема відображує зміст довідкової системи, виконує пошук за ключовим словом, виводить довідку стосовно об'єктів та компонентів, а також містить посилання на адресу сторінок Borland в мережі Інтернет.

### 3. «Гарячі» кнопки

*“Гарячі” кнопки інструментальних панелей* призначені для швидкого запуску найчастіш використовуваних команд головного меню (див. рис. 2.3).



### Рис. Піктографічні кнопки

Клацання лівою кнопкою миші на якій завгодно “гарячій” кнопці спричинить виконання відповідної функції головного меню. (Функції основних “гарячих” кнопок наведені у табл.).

Кнопки піктографічного меню можна редагувати, тобто вилучати й долучати нові кнопки, за допомогою опції Properties з допоміжного меню, котре відкривається за натиснення на цій панелі правої кнопки миші.

Таблиця

#### Функції “гарячих” кнопок

Кнопка	Назва та призначення
	<i>New</i> – створити новий елемент проекту (проект, форму, модуль тощо)
	<i>Open</i> – відкрити файл
	<i>Save</i> – зберегти файл на диску
	<i>Save All</i> – зберегти усі файли проекту на диску
	<i>Open project</i> – відкрити проект
	<i>Add file to project</i> – долучити файл до проекту
	<i>Remove file from project</i> – вилучити файл з проекту
	<i>Help</i> – виклик довідкової системи
	<i>Select unit from list</i> – обрати модуль зі списку модулів, пов’язаних з проектом
	<i>Select form from list</i> – обрати форму зі списку форм, пов’язаних з проектом
	<i>Toggle form/unit</i> – змінити місцями вікно форми й вікно модуля
	<i>New form</i> – створити нову порожню форму й долучити її до проекту
	<i>Run</i> – компілювати і виконати програму
	<i>Pause</i> – зреалізувати паузу в роботі програми
	<i>Trace info</i> – покрокове трасування програми із заходом до підпрограм
	<i>Step over</i> – те саме, але без заходу до підпрограм

## 4. Палітра компонентів

*Палітру компонентів* сформовано у вигляді сукупності сторінок, на кожній з яких розміщуються компоненти, поєднані спільним функціональним призначенням (рис.). Щоб отримати доступ до певного компонента, слід спочатку клацанням миші обрати потрібну сторінку, а вже на цій сторінці обрати потрібний компонент і розмістити його на формі. Компоненти на палітрі подано у вигляді кнопок.



*Рис. Палітра компонентів: а - сторінки Standard; б - сторінки Additional*

Палітру компонентів можна налаштовувати відповідно до вимог розробника. Виклик відповідного діалогового вікна можна здійснити чи то командою *Component / Configure Palette*, чи опцією *Properties* з контекстного меню.





Спочатку розглянемо стандартні компоненти, які дозволяють створювати візуальний інтерфейс у програмних додатках.


## 5. Сторінка Standard






На цій сторінці палітри компонентів містяться базові для Windows елементи, (табл).

*Таблиця*

### Компоненти сторінки Standard

Кнопка	Назва	Призначення
	Frame (Фрейм – рамка)	Окремий контейнер у вигляді окремого вікна для компонентів з можливостями наслідування
	MainMenu (Головне меню)	Панель команд головного меню. Ідентифікатори всіх команд меню визначаються властивістю Items, яка має доступ до кожної команди меню
	PopupMenu (Допоміжне меню)	Контекстне меню для форми чи для іншого компонента, котре зазвичай з'являється у вигляді окремого вікна після натискання правої кнопки миші. Саме для цього кожен компонент має властивість PopupMenu, в якій можна задати посилання на пов'язане з ним меню
	Edit (Однорядковий редактор)	Відображення, введення та виведення лише одного рядка тексту; цей рядок зберігається у властивості Text. Параметри шрифту: назва, розмір, колір, стиль визначається за допомогою властивості Font

Кнопка	Назва	Призначення
	Label (Надпис – позначка)	Надпис на формі, текст якого задається властивістю Caption. Зазвичай використовується для пояснювальних надписів. Властивість AutoSize є ознакою того, що розмір поля визначається його вмістом. Значення true властивості WordWrap надає можливість перенесення тексту надпису по словах. Властивість Alignment визначає спосіб вирівнювання тексту
	Memo (Багаторядковий текстовий редактор)	Багаторядкове текстове вікно для відображення, введення чи виведення тексту та значень даних програми; текст зберігається у властивості Lines
	Button (Кнопка)	Прямокутна кнопка, натискання якої ініціює задані дії у програмі; надпис на кнопці задається властивістю Caption
	CheckBox (Індикатор з прапорцем)	Квадратний індикатор дозволяє користувачу вмикати/вимикати опції програми, оскільки компонент передбачає два стани: "ввімкнено" і "вимкнено" (властивість State)
	RadioButton (Радіокнопка)	Кругла кнопка з двома станами і текстом опису, який специфікує її призначення. Радіокнопки – набір взаємовиключних варіантів вибору: лише одна кнопка може бути обрана на даний момент часу (позначається внутрішнім чорним кружечком), а з попередньо обраної кнопки вибір автоматично знімається. При натисканні радіокнопки властивість Checked змінюється і виникає подія OnClick
	ListBox (Список вибору)	Список текстових варіантів для вибору. Елементи списку містяться у властивості Items, а номер обраного під час виконання програми елемента – у властивості ItemIndex. Можна динамічно долучати, вилучати, вставляти і переміщувати елементи списку за допомогою методів Add, Delete, Insert та Append об'єкта Items. Значення true властивості Sorted встановлює сортування елементів списку за абеткою
	ComboBox (Комбінований список)	Список елементів (властивість Items) розкривається (випадає) для вибору. Значення обраного елемента надається властивості Text. Компонент поєднує в собі два компоненти: ListBox та Edit. Значення true властивості Sorted встановлює сортування елементів списку за абеткою



Кнопка	Назва	Призначення
	GroupBox (Група елементів)	Контейнер, який поєднує на формі групу логічно пов'язаних компонентів
	ScrollBar (Смуга прокручування)	Смуга прокручування з бігунком для керування перегляданням видимої частини форми чи вмісту іншого компонента, приміром для переміщення всередині заданого інтервалу значень певного параметра. Значення властивостей Min та Max встановлюють інтервал припустимих переміщень бігунка. Можна програмно встановлювати бігунк в позицію, задану значенням властивості Position
	RadioGroup (Група радіокнопок)	Є комбінацією групового вікна GroupBox і групи залежних радіокнопок RadioButton. Лише одну кнопку з групи може бути обрано. Перелік кнопок задається властивістю Items. Значення властивості ItemIndex зумовлює, яку саме радіокнопку обрано на цей момент. Можна групувати радіокнопки у кілька стовпчиків за допомогою властивості Columns
	Panel (Панель)	Платформа для розміщення компонентів. Може використовуватися для побудови смуги стану, панелей інструментів
	ActionList (Список дій)	Містить список дій компонентів, таких як пункти меню чи кнопки

## 6. Сторінка Additional

На цій сторінці палітри компонентів розміщено додаткові компоненти, за допомогою яких до програмного додатка можна долучати спеціалізовані інтерфейси елементи Windows (табл.).

Таблиця

### Основні компоненти сторінки Additional

Кнопка	Назва	Призначення
	BitBtn (Кнопка з графікою)	Командна кнопка із зображенням бітового образу (піктограмою) і надписом. Властивість Kind містить набір стандартизованих стилів кнопки з відповідними надписами та графікою: OK, Cancel, Hello тощо. Властивість Glyph дозволяє обрати для зображення на кнопці який завгодно файл формату bmp
	SpeedButton (Кнопка з фіксацією та графікою)	Графічна швидка кнопка для створення у програмі панелей з "гарячих" кнопок

Кнопка	Назва	Призначення
	MaskEdit (Вікно маскованого редагування)	Дозволяє вводити текст (властивість Text) відповідно до заданого шаблону (властивість EditMask)
	StringGrid (Таблиця рядків)	Таблиця для відображення текстової інформації в її рядках та стовпчиках. Властивість Cells дозволяє програмне звертання до комірок компонента. Властивість ColCount задає кількість стовпчиків, а RowCount – кількість рядків
	DrawGrid (Таблиця рисунків)	Таблиця для відображення графічної інформації в її рядках та стовпчиках
	Image (Зображення)	Контейнер для відображення графіки: піктограм, рисунків та метафайлів
	Shape (Фігура)	Прості геометричні фігури – коло та еліпс, квадрат та прямокутник (можна із заокругленими кутами). Вигляд обраної геометричної фігури задається властивістю Shape, а колір та спосіб її забарвлення – двома вкладеними до Brush властивостями: Color та Style
	Bevel (Кромка)	Рамки та лінії з об'ємним виглядом. Властивість Shape визначає вигляд компонента, а значення властивості Style задає опуклу чи угнуту форму об'єкта
	ScrollBar (Вікно з прокручуванням)	Контейнер для відображення зі смугами прокручування за потреби
	CheckBox (Список з індикаторами)	Компонент є комбінацією списку ListBox та індикаторів CheckBox і забезпечує вибір кількох опцій водночас
	Splitter (Відокремлювач)	Відокремлювач використовується для створення панелей зі змінними розмірами для відокремлення компонентів один від одного
	StaticText (Надпис з бордюром)	Надпис є подібний до компонента Label, але забезпечує можливість задавати стиль бордюру. Основна властивість – Caption
	ControlBar (Інструментальна панель)	Застосовується для розміщення компонентів інструментальної панелі
	ApplicationEvents (Події додатка)	Компонент перехоплює події на рівні додатка
	Chart (Діаграми та графіки)	Різноманітні тривимірні діаграми й графіки


#### 4. Додаткові компоненти

Палітри компонентів відрізняються в різних версіях C++ Builder і можуть бути налаштовані відповідно до вимог розробника. Кількість компонентів вимірюється сотнями. (У табл. наведено ще деякі компоненти з різних сторінок палітри компонентів, на які є посилання у наступних розділах).

*Таблиця*

## Деякі додаткові компоненти

Кнопка	Назва	Призначення	Сторінка
	TabControl (Сторінка із закладкою)	Сторінки із закладками, котрі може обирати користувач. Кількість закладок і надписи на них задаються властивістю Tabs. Автоматичне оновлення вмісту сторінок при перемиканні не відбувається, тому програміст має робити це самостійно	Win32
	PageControl (Багатосторінкове вікно)	Сторінки із закладками для економії місця, якими можна керувати. Створення нових сторінок, редагування та вилучення виконується через контекстне меню клацанням правої кнопки миші та обранням відповідної команди. Властивість Caption кожної задає надпис на закладці	Win32
	ImageList (Список зображень)	Список зображень однакового розміру форматів BMP та ICO. Долучення нових зображень і редагування списку зображень здійснюється через відповідне вікно, відкриване подвійним клацанням по компонентіві	Win32
	RichEdit (Вікно редактора у форматі RTF)	Багаторядкове вікно для форматування тексту в форматі RTF, яке дозволяє задавати різні форматування різних фрагментів тексту	Win32
	ProgressBar (Індикатор)	Прямокутний індикатор, який зафарбовується зліва направо заданим кольором у міру виконання певного процесу в програмі. Властивості Min та Max задають інтервал значень індикатора. Властивість Step задає крок змінювання значень позиції індикатора (властивість Position)	Win32
	DateTimePicker (Введення дати і часу)	Календар, який розкривається і дозволяє обирати дату чи час. Властивості Date й Time містять значення дати й часу відповідно. Властивість Kind дозволяє обрати режим роботи чи то з датою, чи з часом	Win32
	MonthCalendar (Введення дати)	Дозволяє обирати дату з календаря. Властивість Date містить обрану дату	Win32

Кнопка	Назва	Призначення	Сторінка
	TrackBar (Полоса з регулятором)	Шкала з позначками й регулятором поточної позиції (варіант лінійки прокручування). Властивості Min та Max задають інтервал значень шкали, причому властивість Position показує поточну позицію регулятора всередині заданого інтервалу. Кількість зображених позначок специфікує властивість Frequency. Те, на скільки позначок має пересунути регулятор, коли користувач клацне мишею на самій шкалі (по обидва боки від регулятора) чи натисне клавіші PageUp та PageDown, задає значення властивості PageSize. Вигляд шкали можна змінювати за допомогою властивостей TickStyle та TickMarks	Win32
	Timer (Таймер)	Використовується для програмування дій, які мають виконуватись з певною періодичністю чи то за заданого відтинку часу. Властивість Interval задає частоту, з якою має відбуватися подія Timer, значення 1000 цієї властивості відповідає одній секунді	System
	OpenDialog (Відкрити файл)	Компонент призначений для виклику діалогового вікна Windows "Відкрити файл". Властивість FileName зберігає ім'я файла, а властивість InitialDir – обраний шлях до теки з файлом	Dialogs
	SaveDialog (Зберегти файл)	Компонент призначений для виклику діалогового вікна "Зберегти файл". Властивості є подібні до компонента OpenFileDialog	Dialogs


## 7. Вікно інспектора об'єктів

*Вікно інспектора об'єктів* (Object Inspector) розташоване ліворуч вікна форми. Інспектор об'єктів забезпечує простий і зручний інтерфейс для змінювання значень властивостей компонентів і керування подіями, на які реагують об'єкти. Інспектор об'єктів має дві сторінки: *Properties* (властивості) і *Events* (події). Властивості компонентів (назва, колір, розмір тощо) задаються на сторінці *Properties*, а події (натискання клавіш миші чи її рух, натискання клавіш клавіатури, вибір пункту меню тощо) - на сторінці *Events* у вікні Object Inspector (рис.). Для того щоб у вікні інспектора об'єктів було відображено атрибути певного об'єкта, слід виокремити цей об'єкт, клацнувши на ньому лівою кнопкою миші.

## 8. Сторінка властивостей Properties

На цій сторінці (рис.) відображаються для виокремленого об'єкта основні властивості та їхні значення.

Значення властивостей можна змінювати. Приміром, якщо клацнути на формі, а тоді обрати її властивість `Caption`, то у вікні зі значенням цієї властивості `Form1` можна ввести новий текст, який і стане заголовком форми.

Властивість `Color` має список можливих значень, який можна відкрити, клацнувши на кнопці  у вікні цієї властивості. Якщо обрати колір `clRed`, то форма чи виокремлений об'єкт набуде червоного кольору. Усю палітру кольорів можна побачити, двічі клацнувши мишею у полі значення кольору.

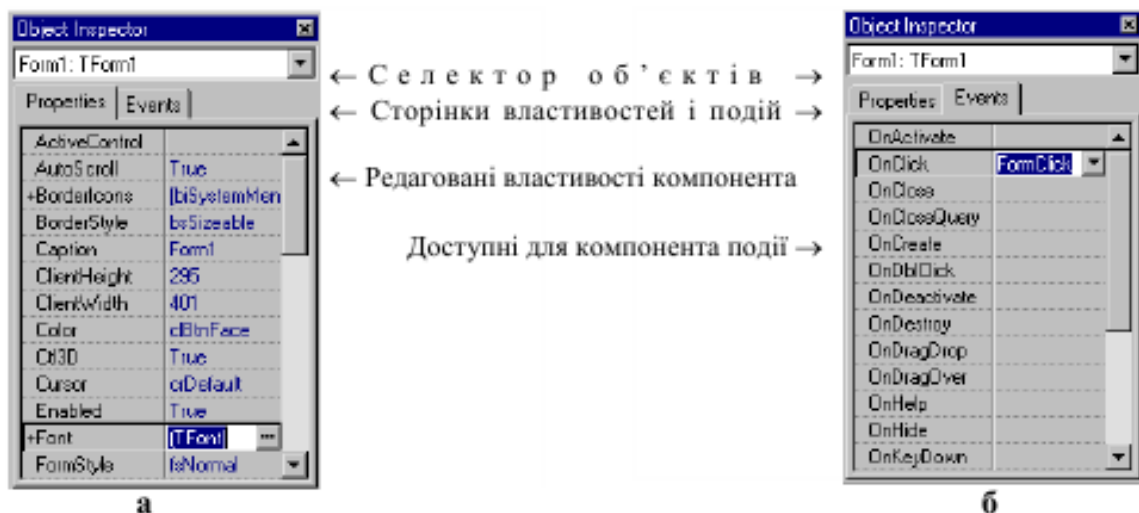



Рис. Вікна інспектора об'єктів: а - сторінка властивостей *Properties*;  
 б - сторінка подій *Events*

Всі властивості мають за замовчуванням певні значення при створенні компонента. Приміром, для компонента `Label` властивість `AutoSize` (Автомасштаб) має значення `true` (істина) без додаткових вказівок. Це означає, що розмір компонента підлаштовується під розмір його вмісту. Але можна задавати потрібний розмір за допомогою властивостей `Width` (довжина) та `Height` (висота), якщо при цьому змінити значення властивості `AutoSize` на `false` (хибність).

Деякі з властивостей мають складну структуру. Клацання миші у полі значень властивості призводить до появи у цьому полі кнопки . Після натиснення цієї кнопки відкриється діалогове вікно, котре і дозволяє встановлювати нові значення цієї властивості. Наприклад, властивість `Font`

дозволяє задавати атрибути тексту: шрифт, розмір, стиль та колір літер.

Якщо на формі треба розмістити кілька однотипних компонентів, то за замовчуванням їхніми іменами буде назва відповідного компонента, наприкінці якого зазначається його порядковий номер на формі. Наприклад, Button1, Button2, ... чи Memo1, Memo2, ...

## 9. Сторінка подій Events

Ця закладка вікна інспектора (рис.) дозволяє визначати реакцію виокремленого компонента на ту чи іншу подію. Найбільш поширеними у застосуванні подіями є такі:

- OnClick - клацання лівою кнопкою миші на об'єкті;
- OnDb1Click - подвійне клацання лівою кнопкою миші на об'єкті;
- OnChange - внесення змін до вмісту об'єкта;
- OnCreate - подія відбувається при створенні форми;
- OnActivate - подія відбувається при активації форми;
- OnMooueDown - натиснення кнопки миші над компонентом;
- OnMooueUp - відпускання кнопки миші над компонентом;
- OnMooueMove - переміщення миші над компонентом;
- OnKeyPre - натиснення символної клавіші клавіатури;
- OnKeyDown - натиснення будь-якої клавіші клавіатури;
- OnKeyUp - відпускання будь-якої клавіші клавіатури;
- OnEnter - подія відбувається в момент набуття компонентом фокуса;
- OnExit - подія відбувається в момент втрати компонентом фокуса;
- OntartDrag - подія відбувається, коли користувач розпочинає перетягування компонента;
- OnDragDrop - подія відбувається в момент відпускання перетягнутого компонента над даним компонентом;
- OnEndDrag - подія відбувається в момент завершення перетягування компонента.

Назви подій наведені у лівій колонці сторінки Events. У правій колонці

подвійним клацанням лівої кнопки миші для виокремленого компонента можна створювати функції відгуку на обрані події, в яких прописати відповідний програмний код. Шаблони цієї функції автоматично відображаються у вікні з програмним кодом, а її прототип автоматично заноситься до відповідного заголовного файлу. Наприклад, для події `OnClick` компонента `Button1`, розміщеного на формі `Form1`, шаблон відповідної функції в `Unit1.cpp` матиме вигляд:

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
}
```

а її заголовок в `Unit1.h`:

```
void __fastcall Button1Click(TObject *Sender);
```

Багато властивостей і методів є притаманні більшості компонентів, приміром: ім'я, розміри, доступність, видимість тощо. (У табл. наведено найчастіш уживані властивості, а в табл. - методи функціонування компонентів).

*Таблиця*

### Основні властивості стандартних компонентів

Властивість	Пояснення	Компоненти з такою властивістю
<code>AutoSize</code>	Автомасштаб: якщо встановити в <code>1</code> ( <code>true</code> ), розмір компонента автоматично адаптується до розміру символів тексту	<code>Label</code> , <code>Edit</code> , <code>Panel</code> , <code>Image</code> та інші
<code>Caption</code>	Надпис	<code>Button</code> , <code>Label</code> та інші
<code>Color</code>	Колір компонента	Більшість видимих
<code>Count</code>	Кількість рядків списку	<code>Memo</code> , <code>ListBox</code>

<b>Властивість</b>	<b>Пояснення</b>	<b>Компоненти з такою властивістю</b>
Enabled	Доступність до компонента. Визначає, чи реагує компонент на події, пов'язані з мишею, клавіатурою і таймером. Значення false блокує доступ, а сам компонент набуває переважно більш світлого кольору	Усі видимі
Font	Параметри шрифту	Button, Label, Edit, Memo та інші
Glyph	Зображення у форматі bmp, яке можна вивести на компонент	BitBtn, SpeedButton
Height	Висота компонента в пікселях	Усі видимі
ItemIndex	Номер обраного елемента списку. Нумерація розпочинається з 0	ListBox
Items	Список можливих значень списку	ListBox, ComboBox, MainMenu та інші
Left	Розміщення лівого краю компонента відносно форми в пікселях. Верхній лівий кут форми має координати 0,0	Усі видимі
Lines	Рядки вікна для введення чи редагування тексту. Нумерація рядків розпочинається з 0	Memo, RichEdit
Name	Ім'я компонента	Усі
ScrollBars	Встановлення у вікні Memo лінійки прокручування. Можливі значення цієї властивості: None, Vertical, Horizontal, Both (відсутня, вертикальна, горизонтальна, обидві)	Memo, StringGrid та інші
Sorted	Якщо встановити в 1 (true), усі елементи списку буде відсортовано за зростанням	ListBox, ComboBox
Text	Текст у вікні редагування	Edit, ComboBox, MaskEdit та інші
Top	Розміщення верхнього краю компонента відносно форми в пікселях. Верхній лівий кут форми має координати 0,0	Усі видимі
Visible	Видимість: якщо встановити в 0 (false), компонент стане невидимим на формі	Усі видимі
Width	Ширина компонента в пікселях	Усі видимі

## Основні методи стандартних компонентів

Метод	Пояснення	Для яких компонентів застосовується
<b>int Add</b> (AnsiString S)	Долучення рядка S до кінця списку. Метод повертає індекс цього рядка. наприклад: int n=Memo1->Lines->Add(S);	Memo, ListBox, RichEdit та інші
<b>Clear()</b>	Очищення вікна	Edit, Memo та інші
<b>Delete</b> (int n)	Вилучення рядка з номером n, наприклад: Memo1->Lines->Delete(n);	Memo, ListBox, RichEdit та інші
<b>Hide()</b>	Приховання компонента	Більшість видимих
<b>Insert</b> (int n, AnsiString S)	Вставлення рядка S у позицію n, наприклад: Memo1->Lines->Insert(n, S);	Memo, ListBox, RichEdit та інші
<b>LoadFromFile</b> (AnsiString fname)	Вивантаження вмісту із зазначеного файлу до поля компонента	Memo, RichEdit та інші
<b>SaveToFile</b> (AnsiString fname)	Зберігання вмісту компонента в файлі із зазначеним ім'ям fname	Memo, RichEdit та інші
<b>SelectAll()</b>	Виокремлення всього тексту	Memo, RichEdit та інші
<b>SetFocus()</b>	Встановлення фокуса (курсору) на цей компонент	Більшість видимих
<b>SetText</b> (char * S)	Долучення тексту S у поточну позицію курсору, наприклад: Memo1->Lines->SetText(S);	Memo та інші
<b>Show()</b>	Показ прихованого компонента	Більшість видимих

## 10 Послідовність створення програмного проекту в C++ BUILDER

Створення програмних проектів у C++ Builder передбачає створення форми, розміщення на ній потрібних компонентів, визначення їхніх властивостей та створення програмних відгуків на задані дії.

Послідовність роботи може бути такою:

- запустити C++ Builder. При запуску автоматично створюється новий проект. Якщо C++ Builder вже працює для створення нового проекту слід з головного меню обрати опцію File/New Application;

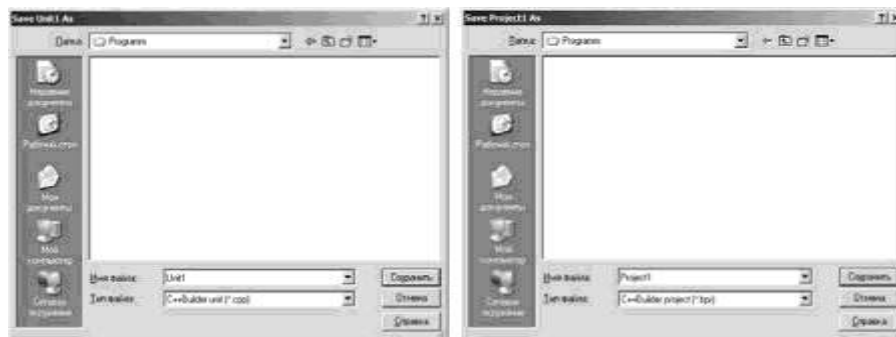
- винести на порожню форму з палітри компонентів усі потрібні компоненти для створення діалогового вікна програми;

- задати на вкладці Properties (властивості) інспектора об'єктів всі потрібні властивості для компонентів, розташованих на формі;

- створити потрібні відгуки на події для розташованих на формі компонентів. Це можна зробити за допомогою вкладки Events (події) інспектора об'єктів чи то безпосередньо з форми, клацнувши двічі по компонентіві. Не можна самостійно прописувати заголовки відгуків чи вилучати автоматично створені;

- у вікні редактора кодів заповнити відгуки на події кодом, а також написати текст усіх потрібних програм. Перехід поміж вікном форми і вікном редактора можна здійснити клавішею <F12>;

- зберегти створений проект за допомогою команди головного меню File/Save Project All (рис.);





*Рис. 2.6. Вікна зберігання файлів проекту*

- запустити проект на виконання одним зі способів: натиснути клавішу <F9>, виконати команду меню Run / Run, натиснути відповідну кнопку на панелі інструментів;

- якщо помилок немає, з'явиться форма без координатної сітки. Якщо ж помилки є, то рядок з помилкою висвічуватиметься червоним кольором, а повідомлення компілятора, розташоване під вікном редактора коду, повідомить, якої саме помилки було припущено. (Характерні помилки компіляції розглянуто пізніше). Згадані помилки слід виправити, зберегти проект і знову запустити проект на виконання;

- провести тестові розрахунки, для чого виконати відповідні дії (вводити

дані, натискати потрібні кнопки тощо). Якщо при виконанні програми були виявлені алгоритмічні помилки їх, зрозуміло, слід виправити. Для довчасного припинення виконання програми, приміром, за її зависання, слід виконати команду Run / Program Reset;

■ доволі часто на практиці при налагодженні програми виникає потреба продивитися проміжкові результати обчислень, приміром за обчислень у циклах сум доданків. Для цього існують спеціальні “гарячі” кнопки покрокового виконання програми (табл.), дію яких можна також відтворити відповідними командами меню Run чи то функціональними клавішами клавіатури <F7> (покрокове трасування програми із заходом до підпрограм ) і <F8> (те саме, але без заходу до підпрограм, - ) . Для переглядання результатів покрокового виконання програми слід відкрити відповідне віконце командою Run / Add Watch. Щоб покрокове виконання програми здійснювати, розпочинаючи з певної команди програми, цю команду слід позначити так званою точкою зупину (Breakpoint) за допомогою команди меню Run / Add Breakpoint чи клавішею <F5>;

■ після налагодження й тестового виконання програми здійснити потрібні обчислення і вийти з C++ Builder. Сеанс завершено.

Рекомендуємо взяти за правило: перед кожним запуском програми обов’язково її зберігати.

### **Хід роботи**

1. Запустіть Borland C++ і познайомтеся з його загальним виглядом.
2. Вивчіть пункти головного меню.
3. Вийдіть із системи Borland C++.
4. Дайте відповіді на контрольні запитання.
5. Зробіть висновок по роботі.

### **Контрольні запитання**

1. З яких двох етапів складається створення програм у C++ Builder?

2. Перелічіть елементи вікна середовища C++ Builder.
3. Для чого потрібна форма проекту?
4. Яке призначення властивостей, подій та методів компонентів?
5. Яке вікно C++ Builder призначене для записування текстів програм?
6. В який спосіб можна здійснювати перемикання поміж вікнами форми й редактора коду?
7. Для чого потрібна палітра компонентів?
8. Перелічіть компоненти, які Ви вже знаєте.
9. Для чого потрібний інспектор об'єктів?
10. В якому вікні задають властивості компонентів?
11. Назвіть основне призначення сторінки подій Events.
12. Перелічіть події, які Ви вже знаєте.
13. Назвіть основні властивості, притаманні більшості компонентів.
14. Вкажіть основні методи стандартних компонентів.
15. З яких основних файлів складається програмний проект у C++ Builder?
16. Яке ім'я за замовчуванням надається головному файлу програми?
17. Назвіть основне призначення заголовного файлу та файлу реалізації.
18. За допомогою якої директиви препроцесора до програми долучають бібліотечні модулі (заголовні файли)?
19. В яких бібліотеках містяться математичні функції?
20. Яке призначення мають фігурні та круглі дужки в C++?
21. Назвіть послідовність створення програмного проекту у C++ Builder.
22. В який спосіб можна зберегти програмний проект?
23. В які способи можна запустити проект на виконання?
24. В які способи можна здійснити перехід поміж вікном форми і вікном редактора коду?