

**МІНІСТЕРСТВО ОСВІТИ УКРАЇНИ
УМАНСЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ПАВЛА ТИЧИНИ**

**ПРОГРАМУВАННЯ НА МОВІ
*VISUAL BASIC***

**Методичні вказівки до курсу
“Інформатика та програмування”**

Умань - 2015

ББК 32.97+22.18

П78

Програмування на мові Visual Basic: Методичні вказівки до курсу “Інформатика і програмування” / Укл. М.В.Дудик, О.В.Малишевський.- Умань: УДПУ, 2015. - 47 с.

ISBN 966-7823-06-7

**Рекомендовано до друку
Вченою радою університету
Протокол №2 від 25.IX.2010 р.**

Рецензенти:

- зав. кафедри теоретичної фізики та інформатики Уманського державного педагогічного університету, проф. А.Н. Кушніренко;
- зав. кафедри математики та інформатики Уманського державної аграрної академії, доц. В.Є. Березовський.

ISBN 966-7823-06-7

© М.В.Дудик, О.В.Малишевський

Зміст

Передмова	3
Лабораторна робота 1. Інтегроване середовище розробки мови програмування Visual Basic 10.0	4
Лабораторна робота 2. Основні поняття та прийоми роботи у VB	9
Лабораторна робота 3. Змінні, оператори і функції	15
Лабораторна робота 4. Керуючі структури	18
Лабораторна робота 5. Цикли у мові VB	21
Лабораторна робота 6. Масиви і списки	23
Лабораторна робота 7. Створення меню	27
Лабораторна робота 8. Використання діалогових вікон	28
Лабораторна робота 9. Спільне діалогове вікно	33
Лабораторна робота 10. Створення підпрограми користувача	37
Лабораторна робота 11. Робота з графікою	40
Лабораторна робота 12. Збереження даних у файлах	43

Передмова

Мова програмування Visual Basic (далі - VB) призначена для розробки програм у середовищі Windows і відноситься до візуальних мов програмування, головною відмінністю яких від традиційних мов програмування є наявність вбудованих у мову елементів інтерфейсу (вікон, меню, кнопок тощо) та можливість візуального формування інтерфейсу майбутньої програми шляхом розміщення у її вікні об'єктів інтерфейсу зі стандартної панелі інструментів.

Даний навчальний посібник містить основні теоретичні відомості про програмування на мові VB та методичні вказівки до виконання лабораторних робіт з практикуму до курсу «Інформатика» для студентів економічного та інших факультетів. За його допомогою читач може швидко познайомитись з базовими поняттями мови і середовища програмування VB, набути під час лабораторного практикуму основні навички створення програм для роботи в середовищі Windows. Крім того, посібник може бути використаний для самостійного навчання програмуванню у середовищі VB. Для більш ефективного виконання лабораторних робіт рекомендується попереднє (домашнє) вивчення теоретичного матеріалу, письмова підготовка відповідей на контрольні запитання, що містяться в кінці опису кожної роботи. Завдання по безпосередній роботі на комп'ютері приведені в методичних вказівках у рамках та виділені курсивом.

ЛАБОРАТОРНА РОБОТА №1. ІНТЕГРОВАНЕ СЕРЕДОВИЩЕ РОЗРОБКИ МОБИ ПРОГРАМУВАННЯ VISUAL BASIC 10.0

1. Запуск VB

Запуск VB здійснюється наступним чином:

- На панелі завдань Windows натисніть кнопку **Пуск**
- В меню **Пуск** виберіть **Програми/Microsoft Visual Basic 10.0**
- В меню **Microsoft Visual Basic 10.0** клацніть на **Visual Basic 10.0**

Після запуску VB з'явиться вікно, де можна створити новий проект. Для створення нового проекту потрібно в стрічці меню обрати команду **Файл-Создать проект-Приложение Windows Forms**.

Завдання. Виконайте запуск Visual Basic 10.0 в режимі створення нового проекту.

2. Інтегроване середовище розробки VB.

Середовище розробки програм на мові VB складається (див. Рис. 1) із стрічки заголовку, стрічки меню, панелей інструментів та кількох вікон. В центрі **VB IDE** розташоване вікно дизайнера форм, в якому виконуються основні дії по створенню програми. Зліва знаходиться панель елементів, що містить піктографічні зображення графічних елементів, з яких будується програма. В правій частині екрану містяться: вікно **Обозреватель решений**, призначене для відображення ієрархічної структури проекту і швидких переходів між його складовими частинами; вікно властивостей **Свойства**, в якому проглядаються і задаються властивості елементів керування; вікно розміщення форми, яке дозволяє за допомогою миші перемістити форму у ту позицію програму, яку вона займатиме при виконанні програми.

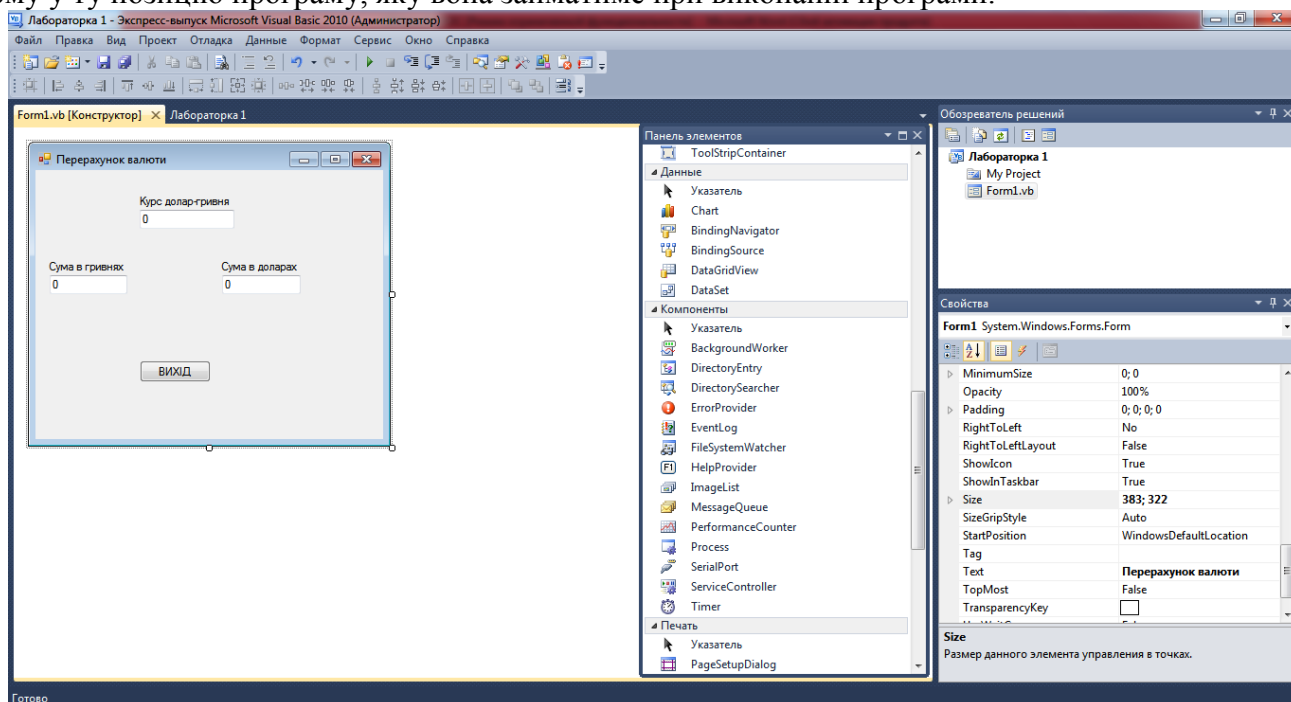


Рис.1. Інтегроване середовище розробки програм Visual Basic 10.0.

2.1. Основні функції **головного меню** представлені наступними пунктами і підпунктами:

- **Файл** - робота з файлами: **Создать проект** (створити **Новий проект**; клавіші швидкого доступу **Ctrl+N**); **Открыть проект** (**Відкрити існуючий проект**; **Ctrl+Q**); **Добавить** (**Создать**, **Существующие проекты**); **Сохранить** (**Зберегти проект під старим іменем**); **Сохранить как** (**Зберегти проект під новим іменем**); **Сохранить <Ім'я_форми.frm >** (**Зберегти форму під старим іменем**; **Ctrl+S**); **Сохранить <Ім'я_форми.frm > как ...** (**Зберегти форму під новим іменем**); **Печать ...** (**Друк форми або проекту**; **Ctrl+P**);

- **Правка** - засоби редагування проекту: **Отменить** (**Відмінити попередню дію користувача** - **Ctrl+Z**); **Вернуть** (**Відновити попередню дію**); **Вырезать** (**Вирізати** - **Ctrl+X**); **Копировать** (**Копіювати**; **Ctrl+C**); **Вставить** (**Вставити**; **Ctrl+V**); **Удалить** (**Видалити**; **Del**); **Выделить все** (**Виділити все**; **Ctrl+A**); **Быстрый поиск** (**Знайти...**; **Ctrl+F**); **Быстрая**

замена (Знайти і Замінити; **Ctrl+H**);

- **Вид** - робота з вікнами середовища розробки **IDE**: **Код** (перехід до вікна **Код**); **Конструктор** (перехід до дизайнера форм; **Shift+F7**); **Начальная страница**; **Другие окна**(**Обозреватель баз данных-Ctrl+Alt+S**; **Список ошибок-Ctrl+W, Ctrl+E**; **Окно свойств-F4**; **Обозреватель решений-Ctrl+Alt+L**; **Панель элементов-Ctrl+Alt+X**; **Веб-обозреватель-Ctrl+Alt+R**; **Результаты поиска**); **Панели инструментов**; **Во весь экран(shift+alt+ввод)**

- **Проект** - містить засоби додавання в проект форм, модулів, файлів тощо;
- **Отладка (Відладка)** - використовується при відладці проекту для пошуку помилок- **Начать отладку (Запуск проекту на виконання; F5)**; (**Запуск проекту на виконання з повною компіляцією; Ctrl+F5**); **Шаг с заходом -F8**; **Шаг с обходом -F9**;

- **Данные** – використовується при перегляді та додаванні нового джерела даних;
- **Формат** – містить засоби вирівнювання, інтервали, порядок та засоби блокування елементів управління;

- **Сервис** - настройка **IDE** та його допоміжні функції: **Диспечер расширений**; **Параметры(Основные, Расширенные, Сброс параметров; Импорт и экспорт параметров); Настройки**;

- **Окно(вікно)** - упорядкування вікон в середовищі розробки: **Плавающая область**; **Создать группу горизонтальных вкладок** (горизонтальне розміщення вікон); **Создать группу вертикальных вкладок** (вертикальне розміщення вікон); **Сброс макета окон**;

- **Справка (Допомога)** - підказки по роботі з **VB**: **Просмотр справки(Ctrl+F1)**; **Управление параметрами справки(Ctrl+Alt+F1)**; **Формулы MSDN**; **Параметр**; **Параметры отзывов пользователей**; **Зарегистрировать продукт**; **Проверить наличие обновлений**; **Техническая поддержка**; **Заявление о конфиденциальности**; **О программе Экспрес выпуск Microsoft Visual Basic -2010**

2.2. Панелі інструментів - це набір маленьких кнопок з піктограмами, натиснення на які замінює виконання певної команди меню. **VB** має у своєму розпорядженні наступні панелі інструментів: **Веб-обозреватель, Конструктор данных, Конструктор запросов, Конструктор представлений, Конструктор таблиц, Макет, Отладка, Построение, Стандартна панель, Схма баз данных, Текстовый редактор, Настройка**. Їх можна відкрити або закрити командою меню **Вид/ Панели инструментов**, яка виводить список панелей, де мишкою потрібно клацнути на стрічці з назвою потрібної панелі. Значок **✓** відповідає розкритим панелям. Список панелей інструментів можна також викликати на екран, клацнувши правою кнопкою миші в будь-якому місці стрічки меню. При необхідності панель інструментів за допомогою мишки можна перетягнути в більш зручне місце екрану і перетворити в "плаваюче" вікно.

При наведенні курсора миші на деяку кнопку панелі інструментів після певної затримки біля неї з'являється ярличок підказки, який пояснює призначення кнопки.

2.3. Панель елементів. Простота розробки програм у **VB** досягається в значній мірі завдяки використанню елементів керування, доступ до яких надає панель елементів. Внутрішні елементи управління вбудовані у **VB** і не можуть бути вилучені з середовища розробки. Це наступні елементи (попередньо зверху вниз перший та другий стовпчики):

- **Стандартные элементы управление: указатель; Button** (при клацанні виникає подія); **CheckBox** (вибір чи видалення параметрів); **CheckedListBox** (список елементів з флаговими кнопками зліва від кожного елемента); **ComboBox**(відображає поле, яке редагують і розкриває список додаткових значень); **DateTimpiker**(дозволяє вибрати дату, час і відобразити їх у вибраному форматі); **Label**(надає елементу управління текст чи інформацію під час виконання); **LinkLabel**(відображає елемент управління «напис», який підтримує форматування, відслідковує і функціонування гіперсилки); **ListBox**(відображає список з якого користувач може вибрати елементи); **ListView**(відображає колекцію елементів в одному з п'яти різних представлень); **MaskedTextBox**(використовує маску для дослідження можливого і неможливого вводу

даних); **MonthCalendar**(відображає місячний календар); **NotifyIcon**(під час виконання відображає значок в області повідомлення, справа від панелі задач); **WebBrowser**(дозволяє переглядати веб – сторінки в середині форми); **NumericUpDown**(відображає одиничне числове значення, яке користувач може збільшити чи зменшити за допомогою кнопок «вверх-вниз» панелі елементів); **PictureBox**(відображає зображення); **ProgressBar**(індикатор виконання операцій); **RadioButton**(дозволяє користувачу вибрати відповідний параметр із групи комбінацій); **RichTextBox**(забезпечує додаткові можливості вводу і редагування тексту); **TextBox**(дозволяє вводити текст і редагувати декілька стрічок і маскувати символи пароля); **ToolTip**(відображає інформацію при наведенні покажчика миші на відповідний елемент управління); **TreeView**(показує користувачу ієрархічну колекцію відмічених об'єктів, які можуть містити зображення); **VScrollBar**(Вертикальна полоса прокрутки); **HScrollBar**(Горизонтальна полоса прокрутки).

- **Контейнери: указатель; FlowLayoutPanel**(обробляє макет компонентів і автоматично компоує їх в макет потоку); **GroupBox**(рамка покругу елементів); **Panel**(дозволяє групувати колекції елементів управління); та інші.
- **Меню и панели инструментов: указатель; ContextMenuStrip**(відображає меню швидкого доступу); **MenuStrip**(відображає команди додатка і параметри згруповані по функціональності); **StatusStrip**(показує інформацію про переглянуті об'єкти, компоненти об'єкта чи операції об'єкта); та інші.
- **Данные: указатель; Chart**(елемент управління діаграмами); **BindingNavigator**(указує інтерфейс користувача для навігації і управління даними); **DateSet**(представляє кеш дані у пам'яті); та інші.
- **Компоненты: Background Worker, DirectoryEntry, DirectorySearcher, ErrorProvider, EventLog, FileSystemWatcher, HelpProvider, ImageList, MessageQueue, PerformanceCounter, Process, SerialPort, ServiceController, Timer.**
- **Печать: указатель, PageSetupDialog, PrintDialog, PrintDocument, PrintPreviewControl, PrintPreviewDialog.**
- **Диалоговые окна: указатель, ColorDialog, FolderBrowserDialog, FontDialog, OpenFileDialog, SaveFileDialog.**
- **Взаимодействие WPF: указатель, ElementHost.**
- **Visual Basic Power Packs: указатель, PrintForm, LineShape, OvalShape, RectangleShape, DateRepeater.**

При наведенні на будь - який елемент управління з вікна панелі елементів покажчика миші, спливає вікно підказка, яка відображає можливості даного елемента.

У VB можливе використання і зовнішніх елементів керування, які існують у вигляді файлів з розширенням .OCX і можуть бути добавлені на панель і вилучені з неї.

Вставка у форму проекту нового елемента керування складається, як правило, з таких кроків: поміщення елемента у форму шляхом подвійного клацання мишкою на його значку в панелі елементів, переміщення елемента у формі в бажане місце, зміна його розмірів та написання коду підпрограми реакції на подію, пов'язану з даним елементом.

2.4. Вікна форми і коду. Вигляд вікна додатку, який воно матиме при виконанні програми, визначається у вікні дизайнера форм, коли програміст добавляє у форму елементи управління. Для кожного вікна дизайнера форм можна відкрити вікно редагування коду, в якому до форм добавляється і редагується код. У вікно редагування коду входять:

- список об'єктів (елементів керування), який надає швидкий доступ до коду, зв'язаному з даним об'єктом;
- список процедур, в якому перераховані події, що пов'язуються з даним елементом керування. Він використовується для швидкого доступу до потрібних подій і процедур;
- бігунок, який ділить вікно редагування коду на два підвікна, що дозволяє бачити одночасно дві частини коду. Він вставляється і вилучається командою **Окно/Разделить**;
- поле для індикаторів - це сіра смужка, в лівій частині вікна редагування коду, де VB поміщує значки закладок і точок зупинки;

- у лівому верхньому куті вікна редагування коду є кнопки перегляду окремої процедури і перегляду всіх процедур модуля, які дозволяють переключатись між режимами перегляду коду проекту.

Команди меню **Окно: Создать группу горизонтальных вкладок; Создать группу вертикальных вкладок** (див. п.1) дозволяють зробити видимими одночасно як форму проекта, так і його код.

2.5. Вікно Обозреватель решений. VB 10.0 дозволяє об'єднувати проекти в групи проектів і зберігати їх у файлі з розширенням **.VBG**. Для керування ними використовується Обозреватель решений, який являє собою ієрархічну деревовидну структуру з проектами на вершині, а компоненти проекту (форми, модулі тощо) йдуть вниз. Для переходу до роботи з будь-якою частиною цього дерева досить клацнути на ній, що показати її. Крім того, вікно Обозревателя решений має під стрічкою заголовку п'ять кнопок вибору режиму перегляду компонента проекту: кнопку виклику редактора коду (**Перейти к коду**), кнопку виклику вікна дизайнера форм (**Открыть в конструкторе**), кнопку переходів між режимами подання компонентів проекта (**Показать все файлы**), **Свойства** і кнопку **Обновить**.

2.6. Вікно властивостей (Свойства). Вікно властивостей дозволяє побачити і змінити початкові установки (значення) елементів керування програми. Під заголовком вікна знаходиться список об'єктів, з якого можна вибрати об'єкт для задання його властивостей. Вікно властивостей має також чотири вкладки: **В алфавитном порядке**, в якій властивості представлені в алфавітному порядку, та **По категориям**, де властивості згруповані по типм, **Свойства** і **События**.

У списку властивостей об'єкту в лівій частині приведені їх імена, а в правій - відповідні значення. При виборі у стрічці імен властивостей певної властивості в нижній частині вікна з'являється її короткий опис.

2.7. Вікно розміщення форми. Вікно розміщення форми дозволяє встановити те положення, яке буде займати форма (або форми) на екрані під час виконання програми. Для цього слід навести в ньому курсор миші на зображення форми (курсор при цьому набуває вигляду чотирьохнапрямленої стрілки) і, утримуючи натиснутою ліву клавішу, перетягнути зображення у потрібну позицію екрана.

Завдання. Ознайомтесь з середовищем розробки VB.

3. Створення додатку в VB

У термінах VB програма називається проектом. Проект включає одну або декілька форм. Під формою розуміють вікно Windows з усіма його атрибутами (заголовком, кнопками розгортання, згортання, виклику керуючого меню тощо). На кожній формі розміщуються як стандартні об'єкти керування (меню, поля введення, прапорці, списки, смуги прокрутки тощо), так і нестандартні, створені будь-яким графічним редактором.

Форми і об'єкти керування мають ряд характеристик, які визначають їх імена, положення, розміри, колір та інші параметри. Об'єкти характеризуються також діями, які над ними можна виконувати, та подіями, що настають в результаті цих дій.

При завантаженні VB в режимі створення нового об'єкту в центрі IDE знаходиться вікно дизайнера форм зі стандартною назвою Form1, яка присвоюється формі по замовчанню. Для вставки у форму нового проекту деякого елемента керування досить двічі клацнути мишкою на відповідній кнопці панелі елементів керування, щоб він з'явився на у центрі форми. Далі його можна перемістити за допомогою миші у інше місце екрану, змінити розміри, "зачепивши" відповідний маркер курсором миші. Аналогічним способом у форму вводяться інші елементи керування.

Для прикладу розглянемо порядок створення проекту, який дозволяв би перераховувати кошти в гривнях у долари згідно з формою:

"сума в доларах"="сума в гривнях"/"курс долар-гривня"

(1)

Зразок форми приведено на рис. 1. Напочатку змінимо заголовок форми. Для цього у вікні **Свойства** потрібно вибрати властивість **Text (Заголовок)** і у рядку введення замість **Form1** набрати нове значення: **"Перерахунок валюти"**. Далі розмістимо у формі три елементи керування **Label**, які служитимуть заголовками полів **"Курс долар-гривня"**, **"Сума в гривнях"** і **"Сума в доларах"**. Щоб знайти цей елемент на панелі елементів, можна скористатись тим, що при наведенні курсора миші на піктограму елемента поруч з'являється надпис про його призначення. Введені елементи по замовчанню отримують заголовки **Label1**, **Label2**, **Label3**, які ми за допомогою властивості **Text** змінимо на **"Курс долар-гривня"**, **"Сума в гривнях"** і **"Сума в доларах"**. Відповідно до довжини цих стрічок змінимо розміри елементів керування **Label**.

Тепер створимо три поля введення: курсу, суми в гривнях, суми в доларах, розмістивши під відповідними заголовками елементи керування - поля введення **TextBox**. По замовчанню в рядках введення знаходяться слова **Text1**, **Text2**, **Text3**; вилучимо їх через властивість **Text**. В нижній частині форми розмістимо елемент управління **Button (Кнопка команди)** і дамо їй заголовок **Вихід** замість присвоєної по замовчанню назви **Command**.

Для полів введення і кнопки виходу встановимо значення характеристики **Name (Ім'я)** у вікні **Свойства** відповідно **Курс**, **Гривня**, **Долар** і **Вихід**. Ці значення характеристики **Name** для кожного об'єкту будуть тим ідентифікатором, по якому вони використовуватимуться в програмі при обчисленнях.

Описаним вище способом у VB без використання жодного оператора створюються вікна інтерфейсу прикладних програм (додатків). Але щоб програма могла виконувати певні операції, її необхідно доповнити підпрограмами реакції на події. Подіями можуть бути введення значень у рядок введення, клацання мишею на кнопці команди і інші. Підпрограми містять код - список інструкцій, написаний на мові VB, які дають програмі вказівки щодо того, що потрібно зробити.

Перейти у режим редагування коду програми можна різними способами: 1) на панелі вікна **Свойства** клацнути кнопкою **Вид Код**; 2) у формі проекту двічі клацнути на елементі керування, в який передбачається вставити код.

Вікно редагування коду містить два списки: об'єктів і процедур. Список об'єктів містить усі об'єкти, що розміщені у формі (для розроблюваної програми - це **Label1**, **Label2**, **Label3** - заголовки, **Курс**, **Гривня**, **Долар** - рядки введення, **Вихід** - кнопка виходу), а також саму форму (об'єкт **Form**) і об'єкт **general**. Останній об'єкт не створювався - він присутній завжди і містить спільні для всіх об'єктів описи змінних і процедури. Якщо у списку об'єктів виділити який-небудь об'єкт, то список процедур міститиме список подій і відповідно процедур реакції на події для виділеного об'єкта.

Повернемося до нашого прикладу. З об'єктами-заголовками тут не пов'язуються жодні події і процедури. Для полів введення реагуватимемо на єдину подію: натискання клавіші для введення чергового символу (процедура **KeyPress**). Ці процедури для об'єктів **Курс** і **Гривня** мають відповідно вигляд:

<pre><i>Private Sub Курс_KeyPress1(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles Курс.KeyPress Select Case Asc(e.KeyChar) Case 46, 48 To 57 Case 13 Call Перерахунок() Case Else e.KeyChar = "" End Select End Sub</i></pre>	<pre><i>Private Sub Гривня_KeyPress1(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles Гривня.KeyPress Select Case Asc(e.KeyChar) Case 46, 48 To 57 Case 13 Call Перерахунок() Case Else e.KeyChar = "" End Select End Sub</i></pre>
---	---

При наборі текстів процедур можна використовувати стандартні засоби редагування

текстів, в т.ч. копіювання:

Описані підпрограми дозволяють вводити у відповідні рядки введення елементів керування цифри від 0 до 9, десяткову крапку. Завершується введення натисканням клавіші **Enter**. Крім того, у рядках введенні допустиме натискання клавішів управління курсором і **Del**.

Для перерахунку суми в доларах за формулою (1) використовується процедура **Перерахунок**, яка має вигляд:

```
Private Sub Перерахунок()
```

```
    Долар.Text = Format(Val(Гривня.Text) / Val(Курс.Text), "#####.##")
```

```
End Sub
```

Для реалізації виходу з програми через кнопку команди **Вихід** при клацанні на ній курсора миші використовується підпрограма:

```
Private Sub Вихід_Click()
```

```
End
```

```
End Sub
```

Завдання. Введіть в програму приведені вище коди процедур. Уважно слідкуйте за помилками, про які VB сигналізує інверсним кольором.

4. Запуск програми на виконання

Запуск програми на виконання здійснюється через меню командою **Отладка/Начать отладку** (клавіша швидкого доступу), або клавіша швидкого доступу **Ctrl/F5**.

Щоб припинити виконання програми, слід виконати команду **Отладка/Остановить** або натиснути на кнопку **X** у стрічці заголовку вікна програми.

Завдання. Виконайте створену перед цим програму. Створіть виконуваний файл на своєму диску.

5. Збереження проекту

Збереження проекту здійснюється командою **Файл/Зберегти**. Якщо проект зберігається вперше, це слід зробити командою **Файл/Зберегти як . . .**, щоб задати його назву. При цьому VB попросить зберегти всі створені за даний сеанс роботи форми і модулі проекту в окремих файлах. VB створить ім'я для кожної форми або модуля, яке співпадає із значенням властивості **Name** відповідної форми або модуля, і добавить підходяще розширення. Але ці імена можна змінити на будь-які інші. Після того, як будуть збережені всі форми і модулі, VB попросить дати ім'я файлу проекту, яке матиме розширення **.VBP**, запропонувавши як варіант ім'я проекту, яке знову можна змінити.

По завершенню роботи з VB і збереження проекту вихід із інтегрованого середовища розробки здійснюється командою **Файл/Выход** (клавіші швидкого доступу **Alt/Q**) або за допомогою кнопки закриття вікна **X** у стрічці заголовку вікна VB.

Завдання. Збережіть на своєму диску створений щойно проект під іменем kurs. Завершіть роботу з VB і Windows.

Контрольні запитання.

1. Як Visual Basic запускається на виконання?
2. Розкажіть про склад інтегрованого середовища розробки VB.
3. Розкажіть про основні пункти і підпункти головного меню VB.
4. Розкажіть про наявні у VB панелі інструментів та їх призначення.
5. Перерахуйте стандартні елементи панелі елементів VB
6. Що входить у вікно коду проекту?
7. Яке призначення вікна властивостей?
8. Як у VB задати положення вікна форми на екрані під час виконання проекту?
9. Як створюється графічний інтерфейс користувача у формі проекту в середовищі розробки VB?
10. Як здійснюється введення коду у VB?
11. Як запустити проект у VB на виконання?
12. Як зберегти проект у VB?

ЛАБОРАТОРНА РОБОТА №2. ОСНОВНІ ПОНЯТТЯ ТА ПРИЙОМИ РОБОТИ У VB

1. Відкриття існуючих проектів

При завантаженні VB користувач має можливість відкрити не обов'язково новий проект, але й такий, що вже існує. Для цього у діалоговому вікні **Файл** замість вкладки **Создать проект** потрібно вибрати вкладку **Открыть** (Існуючий) - для виклику існуючих проектів, або **Последние файлы**(Недавній) - для виклику одного з останніх проектів, з яким велась робота у VB.

Безпосередньо у середовищі VB відкрити проект, з яким недавно велась робота, можна за списком із чотирьох проектів файлів, що відкривались останніми. Вони приведені внизу у меню **Файл**.

Завдання. Відкрийте проект kurs .

2. Властивості, методи, події, процедури

а) Властивості. Властивості використовуються для опису об'єктів і визначають його зовнішній вигляд на екрані та поведінку. Вони можуть змінюватись не тільки під час проектування, але й при виконанні програми, якщо це передбачити відповідним кодом. Кожен об'єкт у VB має свій власний набір властивостей, проте є загальні властивості, притаманні майже всім об'єктам, зокрема:

Властивість	Тип значення	Опис
Location	Integer	Позиція верхнього лівого кута елемента керування відносно лівого верхнього кута його контейнера
Size	Integer	Розмір даного елемента управління в точках (ширина; висота)
TextAlign	Integer	Вирівнювання тексту по полях редагування(Left, Right, Center)
Dock	Integer	Вказує які границі елемента управління прив'язані до контейнера
Name	String	Ім'я, яке використовується для посилань на елемент керування в програмі
Enabled	Boolean	Визначає, чи є елемент керування доступним (значення True), чи ні (False)
Visible	Boolean	Визначає, чи є елемент керування видимим (значення True), чи ні (False)

б) Методи

Методи об'єкта визначають ті задачі, які може виконати даний об'єкт. Метод являє собою сегмент коду програми, впроваджений в об'єкт, завдяки чому об'єкт знає, як виконати ту чи іншу задачу, і немає потреби писати додаткові інструкції. Різні об'єкти мають різні методи, проте існують методи, які притаманні багатьом об'єктам, зокрема:

- **Drag** - оброблює операції "Перетягнути і опустити";
- **Move** - відповідає на запити програми по переміщенню об'єкта;
- **SetFocus** - передає фокус об'єкту, визначеному у виклику метода (фокус - стан, коли об'єкт може сприймати сигнали, які посилає користувач за допомогою миші або клавіатури. В кожен окремих момент часу лише один об'єкт може мати фокус, який виділяється на ньому кольором або пунктирною рамкою і містить, якщо це передбачено, маркер вводу);
- **ZOrder** - визначає спосіб появи об'єкта в контейнері: попереду або позаду інших об'єктів.

в) Події

Крім виконання різних задач, об'єкти програм можуть реагувати на деякі дії користувача чи комп'ютера, наслідком яких має бути настання певних подій. Для обробки подій пишеться відповідний код. Наприклад, такий дії користувача, як клацання кнопкою миші, відповідає подія **Click**, а наслідок цієї події має бути запрограмований у коді.

Кожен елемент керування або форма відповідають за певний набір подій, але зовсім необов'язково обробляти їх усі. Якщо для якоїсь з подій коду не написано, програма буде просто її ігнорувати.

При всій різноманітності подій, на які реагують форми і елементи керування, наступні події є загальними для багатьох об'єктів:

- **Cange** - Користувач змінює текст у полі тексту або полі зі списком;

- **Click** - Користувач клацає на об'єкті основною кнопкою миші;
- **DoubleClick** - Користувач двічі клацає на об'єкті основною кнопкою миші;
- **DragDrop** - Користувач перетаскує елемент керування у нове положення;
- **DragOver** - Користувач перетаскує об'єкт поверх елемента керування;
- **GotFocus** - Об'єкт отримує фокус;
- **KeyDown** - Натискується клавіша при умові, що об'єкт знаходиться у фокусі;
- **KeyPress** - Натискується і відпускається клавіша при умові, що об'єкт знаходиться у фокусі;
- **KeyUp** - Відпускається клавіша при умові, що об'єкт знаходиться у фокусі;
- **LostFocus** - Відбувається безпосередньо перед тим, як об'єкт втрачає фокус;
- **OnMouseDown** - Натискається кнопка миші при умові, що курсор миші знаходиться на об'єкті;
- **OnMouseMove** - Курсор миші рухається поверх об'єкту;
- **OnMouseUp** - Відпускається кнопка миші при умові, що курсор миші знаходиться на на об'єкті.

г) Процедури обробки подій

Реакція програми на певну подію задається процедурами обробки подій, які являють собою набір операторів мови VB з кодом відповіді на подію. Існують три типи процедур: **Sub** - підпрограма, яка не повертає ніяких значень; **Function** - функція, визначена користувачем, по завершенні обчислення якої повертається деяке значення; **Property** - властивість, визначена користувачем. Процедури існують у вигляді блока коду наступної структури:

```
[Public|Private] [Static] Sub|Function|Property Ім'я_функції [(список аргументів)] [As type]
[оператори]
Ім'я_функції=вираз
End Sub|Function|Property
```

(Тут і далі квадратні дужки означають необов'язковий елемент, а вертикальна риска розділяє допустимі елементи із списку, з якого має бути лише один елемент.)

Перша і остання стрічки процедури означають її початок і кінець, між ними - список виконуваних в ній операторів та вираз, який обчислює значення, що повертається у процедуру, яка викликала дану функцію. Ключові слова **Public** і **Private** визначають область дії процедури - у всій програмі чи тільки в модулі, де вона визначена.

Щоб написати код для обробки певної події, генерованої конкретним елементом керування, потрібно перейти у вікно редагування коду (Правою кнопкою миші **Перейти к коду** або пункт **Код** у меню **Вид**) і вибрати об'єкт та подію, для якої має бути створено код. VB автоматично створить блок коду процедури обробки події, який являє собою заготовку процедури з іменем процедури на початку і оператором **End Sub** у кінці, Ім'я процедури обробки події у відповідності з домовленістю про імена містить ім'я об'єкта та ім'я події, відділені рисою підкреслення. Коли заготовка процедури буде готова, в неї записуються всі оператори, що визначатимуть реакцію програми на настання події.

Завдання. Викличте до розгляду код проекту kurs і проаналізуйте наявні в ньому події та відповідні процедури.

3. Параметри вікон

Для того, щоб форма проекту виглядала привабливо, необхідно задати їй та її елементам керування належні розміри, положення і стиль.

Розміри об'єкта (форми або її елементів) під час проектування можна змінити одним з наступних способів:

1) за допомогою миші, перетягуючи маркери розміру (вузли редагування), що з'являються навколо форми чи виділеного елемента у вікні дизайнера форм;

2) змінивши значення властивості **Size** у вікні властивостей, яка задає ширину і висоту об'єкта через крапку з комою або у властивості **MinimumSize** її компоненти **Width**, **Height**. Властивість - це кількісно змінюваний атрибут об'єкту. Для зміни властивостей слід виділити об'єкт у формі, перейти у вікно властивостей, вибрати вкладинку **В алфавитном порядку**, у списку властивостей за допомогою смужки прокрутки знайти потрібну властивість, двічі

клацнути справа від неї і ввести нове числове значення через крапку з комою. Одиницею вимірювання відстаней і розмірів об'єктів на екрані по замовчанню є твіп - 1/1440 дюйма. При заданні розмірів і положення об'єктів у твіпах їх вигляд на екрані не залежатиме від типу монітора.

Змінити положення форми на екрані або елемента керування у формі під час проектування також можна кількома способами:

1) за допомогою маніпулятора миша, перетягуючи виділений об'єкт у потрібне місце вікна (для зміни положення форми на екрані використовується вікно **Form Layout**);

2) змінивши значення властивостей **Left** і **Top** у вікні властивостей, які задають відстань від об'єкта до лівого і верхнього краю форми (екрану для форми) відповідно.

VB має також можливість змінювати описані властивості програмно, за допомогою коду.

Щоб придати формі належного професійного вигляду, необхідно задати їй належний стиль за допомогою властивості **FormBorderStyle**. У VB є 7 стилів:

Значення	Стиль	Опис
0	None	Без рамки, без стрічки заголовку і без можливості переміщення вікна
1	Fixed Single	Не допускає зміну розмірів вікна за допомогою перетаскування його сторін, але допускає використання кнопок згортання і розгортання вікон
2	Fixed3D	Допускає зміни розмірів вікна, стрічка заголовку виділена.
3	Sizable	Допускає зміну розмірів
4	Fixed Dialog	Не допускає можливості переміщення, мінімізації або максимізації вікна
5	Fixed ToolWindow	Схожий на Fixed Dialog, але стрічка заголовка вікна більш коротка і має більш мілкий шрифт
6	Sizable ToolWindow	Схожий на Fixed ToolWindow, але допускає зміну розмірів вікна за допомогою перетаскування його сторін

4. Робота з елементами керування

Вставити новий елемент керування у форму з панелі елементів можна не лише подвійною фіксацією курсора миші на відповідній кнопці, але й за допомогою виділення і перетаскування. При останньому способі потрібно: один раз клацнути мишкою на потрібній кнопці панелі елементів, перейти у форму, натиснувши ліву кнопку і, утримуючи її, виділити у формі прямокутну область, куди буде вставлено об'єкт, а потім відпустити кнопку: елемент керування буде вставлено у форму.

Щоб вилучити деякий елемент керування із форми, його слід спочатку виділи у формі, клацнувши один раз на ньому мишкою, а потім натиснути клавішу **Del**.

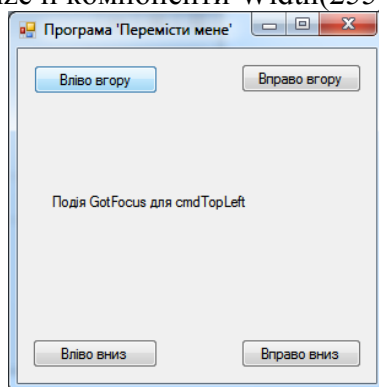
При внесенні елемента керування у форму VB присвоює йому ім'я по замовчанню, проте рекомендується змінювати його так, щоб воно несло більше інформації про призначення елемента. Серед програмістів на VB поширене використання імен типу `ctlName`, де `ctl` - префікс, який визначає тип елемента, `Name` - інформативне з точки зору функціонального призначення ім'я елемента керування. Так, до імені прапорця перевірки (`CheckBox`) прийнято добавляти префікс `chk`, до поля зі списком (`ComboBox`) - `cbo`, до кнопки команди (`CommandButton`) - `cmd`, до загального діалогового вкна (`CommonDialog`) - `cdl`, до вікна списку каталогів (`DirListBox`) - `dir`, до вікна списку накопичувачів (`DriveListBox`) - `drv`, до вікна списку файлів (`FileListBox`) - `fil`, до форми (`Form`) - `frm`, до рамки (`Frame`) - `fra`, до сітки (`Grid`) - `grd`, до горизонтальної чи вертикальної полоси прокрутки (`HScrollBar`, `VScrollBar`) - `hsb`, `vsb`, до зображення (`Image`) - `img`, до надпису (`Label`) - `lbl`, до списку (`ListBox`) - `lst`, до меню (`Menu`) - `mnu`, до перемикача (`OptionButton`) - `opt`, до поля тексту (`TextBox`) - `txt` та інші.

Завдання. Створіть у VB проект Password, форма якого має заголовок Перевірка паролю, прапорець перевірки, поле тексту і надпис до нього "Введіть пароль", а також кнопку команди із заголовком "Кінець вводу". Змінюючи розміри і положення на екрані форми проекту, розміри і положення у формі її елементів керування, підберіть найвдаліші (на Ваш погляд). Змінюючи значення властивості FormBorderStyle, перегляньте відповідні їм

вигляди форми. Дайте всім об'єктам проекту підходящі імена, використовуючи властивість Name у вікні властивостей. Збережіть проект на своїй дискеті під іменем Password.

5. Приклад використання властивостей, методів і подій

Створимо новий проект, який демонструє переміщення форми на екрані у певному напрямку в результаті натискання користувачем відповідних кнопок команди (див. рис.). Для цього спочатку необхідно створити нову форму з чотирма кнопками команд по кутам і надпис у центрі, який повідомлятиме про переміщення миші і про те, який з елементів керування знаходиться у фокусі. Змінимо ім'я форми проекту Form1 на frmMain та встановимо значення властивості FormBorderStyle форми рівним 1 - Fixed Single. Властивості Name кнопок команд присвоїмо значення cmdTopLeft (вгору-вліво), cmdTopRight (вгору-вправо), cmdBottomLeft (вниз-вліво), cmdBottomRight (вниз-вправо). Для надпису задамо значення властивості Name - lblStatus, властивості MinimumSize її компоненти Width(255), Height(30) або ж Size(255;30).



В порядку експерименту надписи на кнопках (властивість Caption) задамо в програмно в спеціальній процедурі Form_Load(), для чого двічі клацнемо у будь-якому місці форми frmMain, крім заголовка, щоб викликати вікно редагування коду, і введемо в процедуру Form_Load() приведенний нижче код (знак апострофа ' у коді починає коментар до програми; як невиконуваний оператор коментар при вводі коду можна опустити):

```
Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
' Задання надписів на кнопках команд
```

```
cmdTopLeft.Text = "Вліво вгору"
```

```
cmdBottomLeft.Text = "Вліво вниз"
```

```
cmdTopRight.Text = "Вправо вгору"
```

```
jjkj.Text = "Вправо вниз"
```

```
' Очистка поля з інформацією про стан програми
```

```
lblStatus.Text = ""
```

```
' Зміна тексту стрічки заголовка форми так, щоб він повідомляв назву додатка
```

```
Me.Text = "Програма 'Перемісти мене'"
```

```
End Sub
```

Описаний вище процес задання початкових значень властивостей називається ініціалізацією. Створимо тепер код для переміщення форми на екрані, який буде добавлено у процедури обробки події Click() для всіх кнопок команд:

```
Private Sub cmdTopLeft_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdTopLeft.Click
```

```
Me.Top = 0
```

```
Me.Left = 0
```

```
End Sub
```

```
Private Sub cmdTopRight_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdTopRight.Click
```

```
Me.Top = 0
```

```
Me.Left = Screen.PrimaryScreen.Bounds.Width - Me.Width
```

```
End Sub
```

```
Private Sub cmdBottomRight_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles jkj.Click
```

```
Me.Top = Screen.PrimaryScreen.Bounds.Height - Me.Height
```

```
Me.Left = Screen.PrimaryScreen.Bounds.Width - Me.Width
```

```
End Sub
```

```
Private Sub cmdBottomLeft_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdBottomLeft.Click
```

```
Me.Top = Screen.PrimaryScreen.Bounds.Height - Me.Height
```

```
Me.Left = 0
```

```
End Sub
```

У даному фрагменті коду є посилання на об'єкт **Screen**, який означає екран монітора.

Приведемо також коди процедур, які інформували б користувача про настання певних подій, а саме про події **MouseDown** і **MouseUp** для форми та про подію **GotFocus** для кожної з кнопок. Цю інформацію програма повинна поміщати в елемент керування **lblStatus**. Те саме відбуватиметься, якщо для переходу від однієї кнопки команди до іншої використовуватиметься клавіша **Tab**.

Для відкриття вікна редагування коду процедури обробки події **MouseDown** потрібно двічі клацнути на формі і у вікні редагування коду об'єкта **Form** вибрати подію **MouseDown**. З'явиться заготовка процедури, яку слід доповнити приведеним нижче кодом. Потім, не виходячи з вікна редагування коду об'єкта **Form**, виберіть подію **MouseUp** і у заготовку її процедури введіть потрібний код.

```
Private Sub frmMain_MouseDown(ByVal sender As Object, ByVal e As System.Windows.Forms.MouseEventArgs) Handles Me.MouseDown
```

```
lblStatus.Text = "Подія MouseDown"
```

```
End Sub
```

```
Private Sub frmMain_MouseUp(ByVal sender As Object, ByVal e As System.Windows.Forms.MouseEventArgs) Handles Me.MouseUp
```

```
lblStatus.Text = "Подія MouseUp"
```

```
End Sub
```

Доповнимо також проект процедурами обробки події для виводу повідомлень про те, яка з кнопок команд отримала фокус. Для цього при перебуванні у вікні редагування слід вибрати відповідний об'єкт, наприклад кнопку команди **cmdBottomLeft**, і процедуру **GotFocus** та у заготовку процедури ввести код. Коди для всіх кнопок приведені нижче:

```
Private Sub cmdBottomLeft_GotFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles cmdBottomLeft.GotFocus
```

```
lblStatus.Text = "Подія GotFocus для cmdBottomLeft"
```

```
End Sub
```

```
Private Sub cmdTopRight_GotFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles cmdTopRight.GotFocus
```

```
lblStatus.Text = "Подія GotFocus для cmdTopRight"
```

```
End Sub
```

```
Private Sub cmdTopLeft_GotFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles cmdTopLeft.GotFocus
```

```
lblStatus.Text = "Подія GotFocus для cmdTopLeft"
```

```
End Sub
```

```
Private Sub cmdBottomRight_GotFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles jkj.GotFocus
```

```
lblStatus.Text = "Подія GotFocus для cmdBottomRight"
```

```
End Sub
```

Завершивши і відредагувавши код проекту, можна запустити проект на виконання.

Завдання. Виконайте описаний вище приклад і збережіть його на своєму диску під іменем **MoveMe і вийдіть з VB.**

Контрольні запитання.

1. Як у VB відкриваються існуючі проекти?
2. Яке призначення властивостей? Які загальні властивості Ви знаєте?
3. Яке призначення методів? Які загальні методи Ви знаєте?
4. Що розуміють під подіями? Які загальні події Ви знаєте?
5. Що означає термін "фокус" у VB?
6. Що розуміють під терміном процедура обробки подій?
7. Який синтаксис процедур у VB?
8. Як вводиться код процедури у проект?
9. Які параметри мають вікна об'єктів у проектах VB та як їх можна змінити?
10. Які стилі можуть мати об'єкти у VB і чим вони відрізняються?