

Міністерство освіти і науки України
Вищий державний навчальний заклад
“НАЦІОНАЛЬНИЙ ГІРНИЧИЙ УНІВЕРСИТЕТ”



МЕХАНІКО–МАШИНОБУДІВНИЙ ФАКУЛЬТЕТ
Кафедра основ конструювання механізмів і машин

ПРАКТИКУМ
до виконання лабораторних робіт з дисципліни "Комп'ютерна техніка та програмування" (розділ - Програмування в середовищі Visual Basic) для студентів напрямку 0701 "Транспортні технології"

Дніпропетровськ НГУ
2011

УДК 681.3.06
ББК 32.973.26-018.1 Visual Basic
С12

Практикум до виконання лабораторних робіт з дисципліни "Комп'ютерна техніка та програмування" для студентів напрямку 0701 "Транспортні технології" / М.О. Дудко, С.В. Балашов, Т.О. Письменкова, С.В. Розов. Дніпропетровськ: Вищий державний навчальний заклад "НАЦІОНАЛЬНИЙ ГІРНИЧИЙ УНІВЕРСИТЕТ", 2011. - 56 с. Бібліографія: 10 назв.

У практикумі розглянуті принципи візуального програмування на мові Visual Basic. На конкретних прикладах показані можливості середовища розробки, методика створення програми і особливості процесу програмування. Описується застосування Visual Basic до реалізації лінійних алгоритмів, розгалужених і циклічних. Кожна з тих тем, що розглядаються, супроводиться коротким теоретичним вступом, що пояснює приведені приклади.

Мета рекомендацій - надання допомоги студентам при вивченні курсу "Комп'ютерна техніка та програмування", а також при виконанні індивідуальних завдань.

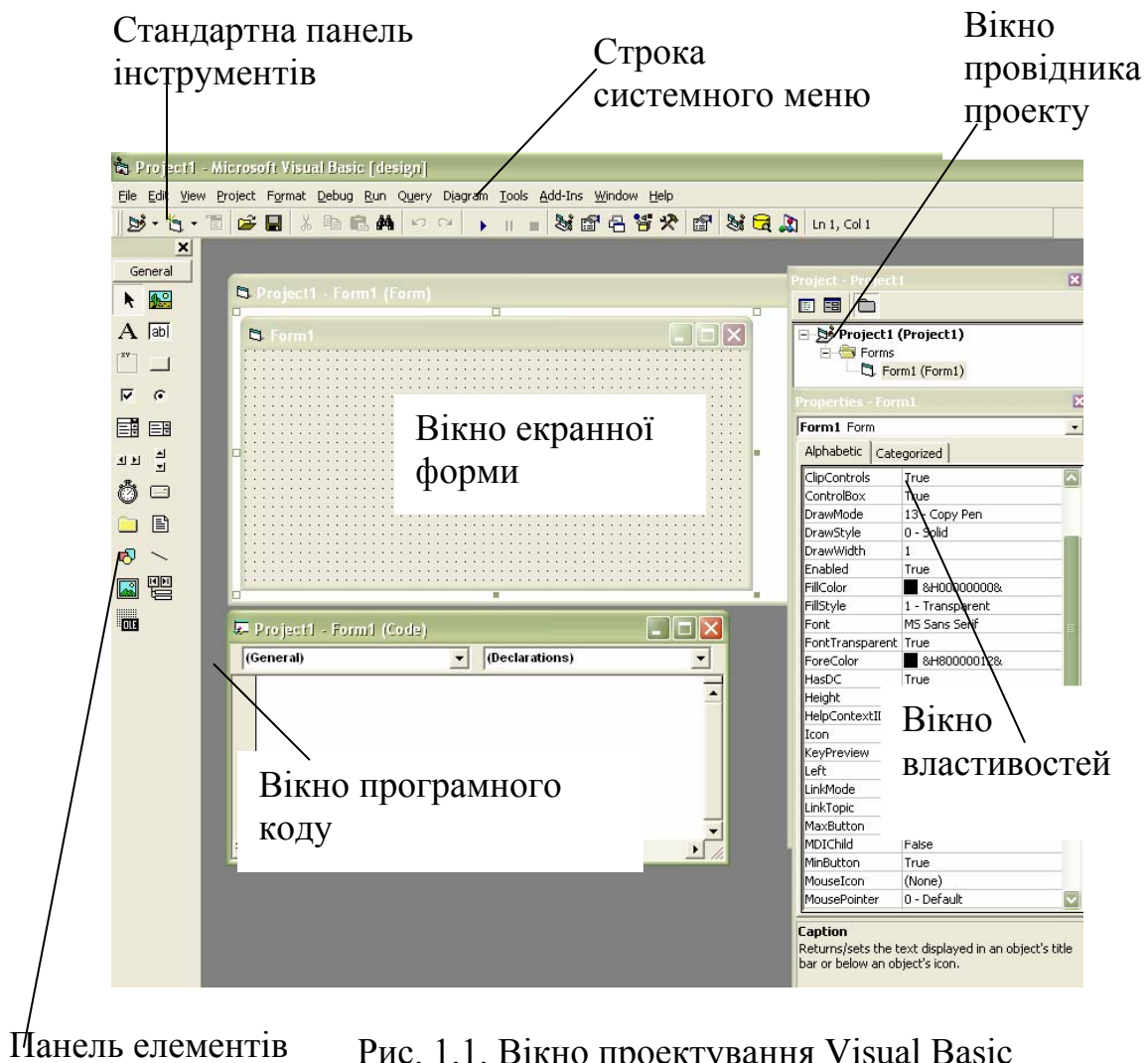
Допомога призначена для студентів напрямку 0701 "Транспортні технології", проте може бути використано студентами інших спеціальностей.

ЗМІСТ

1. Основні поняття об'єктно-орієнтованого програмування в Visual Basic..	4
2. Основні принципи розробки програм (проектів) у середовищі Visual Basic.....	9
3. Лабораторна робота № 1.....	16
4. Лабораторная работа №2.....	20
5. Лабораторна робота № 3.....	24
6. Лабораторна робота № 4	28
7. Лабораторна робота № 5.....	33
8. Лабораторна робота № 6	39
9. Лабораторна робота № 7.....	44
10. Лабораторна робота № 8.....	50
Література.....	58

1. Основні поняття об'єктно-орієнтованого програмування в Visual Basic

1. Загальні відомості. Технологія роботи у середовищі Visual Basic базується на ідеях об'єктно-орієнтованого та візуального програмування. Ідея об'єктно-орієнтованого програмування полягає в інкапсуляції (об'єднанні) даних і засобів їх опрацювання (методів) у тип, який називається об'єктом. Прикладами об'єктів можуть бути елементи керування у вікні: кнопки, списки, текстові поля тощо. Середовище візуального програмування Visual Basic це графічна автоматизована оболонка над об'єктно-орієнтованою версією мови Basic (рис. 1.1). Якщо у мові Basic структурними одиницями є дані та команди, то тут такою структурною одиницею є візуальний об'єкт, який називається компонентом. Автоматизація програмування досягається завдяки можливості переносити компонент на форму (у програму) з палітри компонентів і змінювати його властивості, не вносячи вручну змін до програмного коду.



Формою називають компонент, який володіє властивостями вікна Windows і призначений для розташування інших компонентів. Компоненти служать для

організації діалогу з користувачем. Це кнопки, списки, текстові поля, зображення, конструктор мошо тощо. Вони відображаються на екрані під час виконання програми.

Проект-це сукупність файлів, з яких складається програма, створена в середовищі Visual Basic.

2. Інструменти середовища Visual Basic. Вікно середовища містить головне меню, панель інструментів, а також:

- палітру компонентів (**ToolBox**);
- вікно властивостей об'єктів (**Properties Window**);
- вікно форми (**Form**);
- редактор коду (**Code**).

Усі ці засоби можна відкрити у разі потреби командами головного меню **View** → **ToolBox**, **View** → **Properties Window**, **View** → **Form** та **View** → **Code**.

3. Головне меню та панель інструментів. Головне меню складається з таких команд (пунктів): **File**, **Edit**, **View**, **Project**, **Format**, **Debug**, **Run**, **Query**, **Diagram**, **Tools**, **Add-Ins**, **Window**, **Help**.

Меню **File** містить стандартні команди, призначені для роботи з файлами проекту. За допомогою цих команд можна створити новий проект (**New Project**), відкрити чи закрити файл проекту (**Open Project**, **Remove Project**), додати проект (**Add Project**), зберегти проект чи форму (**Save Project**, **Save Project As**, **Save Form...**, **Save Form.. As**), роздрукувати вибрані файли проекту (**Print**) чи створити ехе-модуль (**Make Project.exe**).

Меню **Edit** містить стандартні команди для пошуку та заміни фрагмента тексту (**Find**, **Replace**, **Find Next**), команди роботи з буфером (**Copy**, **Cut**, **Paste** тощо) тощо.

У меню **View** знаходяться команди візуалізації елементів середовища.

Меню **Project** містить команди керування проектом, зокрема команди додавання файлів до проекту (**Add Form**, **Add Module** тощо).

За допомогою команд меню **Format** можна вирівнювати компоненти відносно сітки та між собою (**Align**), задавати порядок відображення компонентів, які перетинаються (**Order** → **Bring to Front**, **Order** → **Send to Back**), змінювати розміри вибраного компонента (**Make Same Size**) і т.і.

Меню **Debug** призначене для налагодження проекту та пошуку помилок.

Меню **Run** містить команди керування роботою програми (**Run**, **Break**, **End**, тощо).

У меню **Tools** знаходяться команди для задання параметрів середовища.

Панель інструментів служить для розташування кнопок інструментів. На ній можуть міститися кнопки всіх згаданих команд.

4. Палітра компонентів. Палітра компонентів розташована в окремому вікні (рис. 1.2). Щоб помістити компонент у центр вікна форми, двічі клацають на його піктограмі. Якщо потрібно розташувати компонент десь на формі, клацають один раз на його піктограмі і обводять контур у потрібному місці форми. Вибраний компонент можна переміщати на формі, а також змінювати розміри, перетягуючи його маркери.

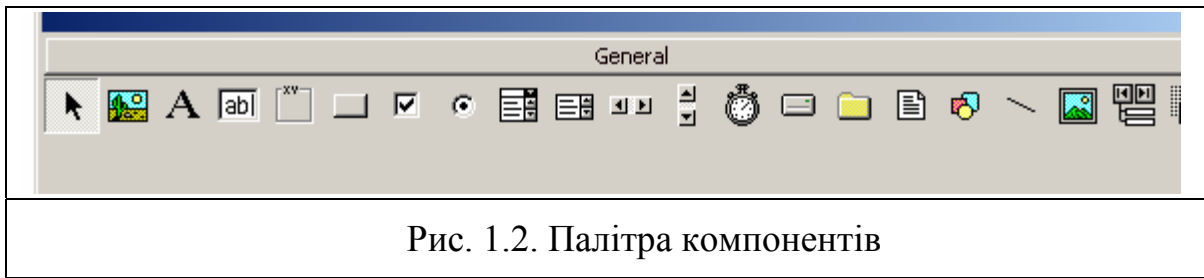


Рис. 1.2. Палітра компонентів

5. Вікно властивостей об'єктів. За допомогою вікна властивостей можна задавати початкові значення властивостей об'єкта. Вікно інспектора об'єктів містить список компонентів поточної форми, а також дві закладки для властивостей: упорядковані за алфавітом (**Alphabetic**) та за призначенням (**Categorized**). Щоб активізувати вікно інспектора об'єктів, використовують клавішу **F4**. Список властивостей складається з двох стовпців: лівий містить назви властивостей компонентів, а правий — їхні значення (рис. 1.3).

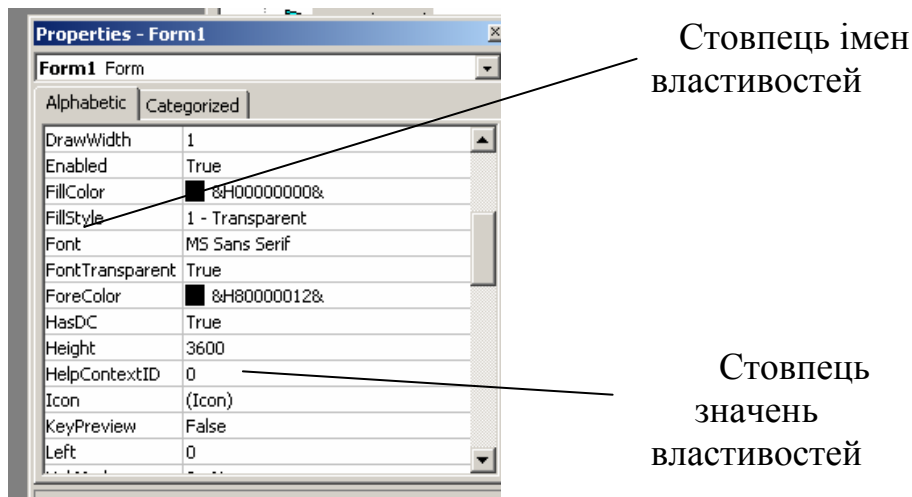


Рис. 1.3. Вікно властивостей об'єктів

Для введення значень властивостей числового та текстового типу (**Width**, **Name** тощо) використовується стандартне поле введення. Значення властивостей перерахованого типу (**Alignment**, **Mouse-Pointer** тощо) задаються комбінованим списком, звідки вибирають потрібне. Деякі комплексні властивості (**Font**, **Picture**, **Icon** тощо) використовують діалогові вікна, набір керуючих елементів яких залежить від конкретної властивості.

6. Вікно форми. Форма — це вікно Windows, яке утворюється в одному з можливих для вікон стилів (рис. 1.4). Увесь внутрішній простір є робочою областю, яка має сітку вирівнювання для зручного розташування компонентів на формі. Для виконання групових операцій декілька компонентів можна об'єднувати. Для цього необхідно натиснути на ліву клавішу миші і переміщенням вказівника охопити всі потрібні компоненти. У групу долучають компоненти, які хоча б частково попадають в охоплену область. Можна також долучити/вилучити окремий елемент. Для цього необхідно натиснути клавішу **Shift** та, не відпускаючи її, вибрати мишею потрібний компонент на формі. Вилучення виокремлених компонентів чи групи виконується клавішею **Delete**.

Переміщення виокремленого компонента в межах форми здійснюється мишею. Над компонентами та їхніми групами можна виконувати операції вирізування, копіювання в буфер обміну і вставляння з буфера.

Вирівнювати компоненти можна як відносно вікна форми, так і один відносно одного. Для цього використовується команда головного меню **Format** → **Align**. Інша можливість — безпосередньо задати властивості **Left** та **Top** компонентів. Компоненти у групі вирівнюються відносно того компонента, який попав у групу першим.

7. Структура проекту. Проектом називають сукупність файлів, з яких Visual Basic створює готову для виконання програму. До складу кожного проекту входять наступні файли:

- файл проекту ***.vbp**. Це невеликий файл, який містить посилання на всі файли проекту та ініціалізує програму;
- файли опису всіх форм, які входять у проект: файл модуля ***.bas** і файл форми ***.frm**. У цих файлах містяться тексти Basic - програми;
- файл ресурсів програми ***.frx**. У ньому описані ресурси, які не належать формі, наприклад, піктограма програми;
- файл параметрів проекту ***.vbw**;

Для збереження проекту необхідно задати імена форм (автоматично пропонуються імена **Form1.frm**, **Form2.frm**, ...), модулів (**Module1.bas**, **Module2.bas**, ...) та ім'я проекту (**Project1.vbp**). Ці імена можна змінити на власні. Для переміщення проекту на інший комп'ютер необхідно мати лише файли таких типів: ***.vbp**, ***.frm**, ***.bas**, ***.frx**. Інші файли створюються автоматично.

8. Редактор коду. Редактор коду програми знаходиться в окремому вікні (рис. 1.5). У верхній частині цього вікна розміщено два комбіновані списки компонентів форми та подій, які можуть бути до них застосовані. Застосування події до певного об'єкта веде до появи заготовки базового коду відповідної процедури (підпрограми) у вікні редактора. Заготовка (шаблон) складається з заголовка процедури **Private Sub <ім'я об'єкта>_<ім'я методу> ()** та ключових слів **End Sub**.

Зазначимо, що у комбінованому списку об'єктів є засіб (**General**), який після вибору у правому списку елемента (**Declarations**) дає програмістові доступ до секції опису глобальних змінних. Заготовку власної функції можна вставити у код програми за допомогою команди головного меню **Tools** → **Add Procedure**. Доступ до такої функції здійснюється вибором у лівому верхньому списку редактора коду позиції (**General**), а у правому - назви цієї функції. Крім

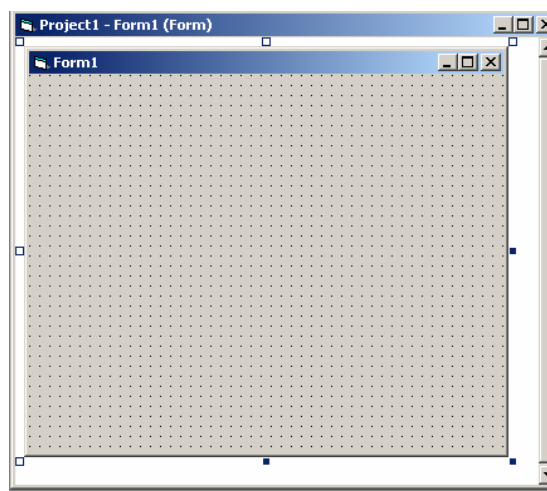


Рис. 1.4. Вікно форми

того, відкрити текст будь-якої процедури чи функції можна і безпосередньо у вікні коду за допомогою клавіш зі стрілками чи смуг прокручування

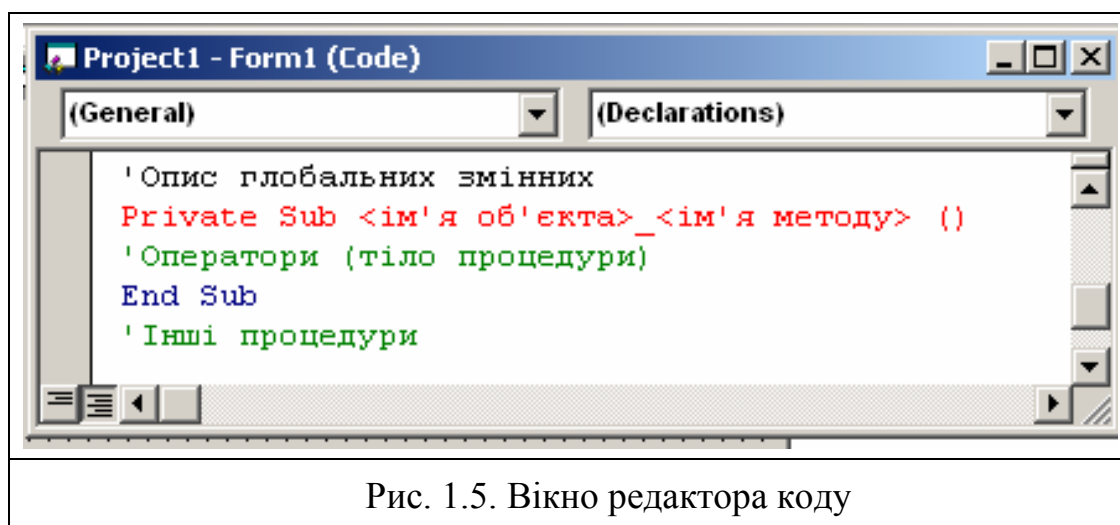


Рис. 1.5. Вікно редактора коду

2. Основні принципи розробки програм (проектів) у середовищі Visual Basic

2.1. Загальні відомості

Алгоритмічна мова Visual Basic (VB) - мова об'єктно-орієнтованого проектування. Це значить, що в ньому можна маніпулювати готовими об'єктами й методами їхньої обробки на рівні операторів алгоритмічної мови.

В об'єктно - орієнтованій мові використовуються такі поняття як:

- об'єкти;
- властивості;
- події;
- методи.

2.2. Об'єкти





Об'єкти в Visual Basic - це пристрої або загальні елементи Windows-додатків, використовувані більшістю створюваних програм. При об'єктно-орієнтованому програмуванні практично всі компоненти комп'ютерної системи є об'єктами. У Visual Basic виділяються такі групи об'єктів:

- *глобальні (global objects):* **clipboard** (буфер обміну), **debug** (налагодженик), **printer** (принтер), **screen** (екран), **app** (додатки);
- *інтерфейсні або об'єкти взаємодії:* **form** (екранна форма), **controls** (керуючі елементи, наприклад, ті, котрі присутні на панелі інструментів);
- *об'єкти бази даних.*

У першу чергу будуть використовуватися об'єкти вікна панелі елементів управління (табл. 2.1).

Таблиця 2.1

Опис вбудованих елементів управління

Елемент	Назва	Призначення
	Pointer (Показчик)	Вибір елементів управління
	PictureBox (Картинка)	Виведення графічних елементів у формі (може використовуватися як контейнер)
Елемент	Назва	Призначення
	Label (Мітка)	Відображення написів у формі
	TextBox (Текст)	Введення тексту

	Frame (Рамка)	Об'єднання в групу різних елементів управління (такі об'єкти називаються <i>контейнерами</i>)
	Command Button (Кнопка, що управляє)	Виконання додатком певних дій, викликаних натисненням на кнопку, що управляє
	Checkbox (Прапорець)	Установка/відключення налаштувань
	OptionButton (Перемикач)	Вибір користувачем одного з декількох можливих пунктів (їх повинно бути не ' менше двох)
	ComboBox (Комбінований список)	Вибір елемента із списку, що розкривається
	ListBox (Список елементів)	Вибір користувачем якого-небудь елемента з тих, які є в списку
	HscrollBar (Горизонтальна смуга прокрутки)	Перегортування в горизонтальному напрямі наявного в іншому елементі управління списку
	VscrollBar (Вертикальна смуга прокрутки)	Перегортування у вертикальному напрямі наявного в іншому елементі управління списку
	Timer (Таймер)	Виконання додатком дій в реальному часі
Елемент	Назва	Призначення
	DriveListBox (Список дисків)	Вибір користувачем якого-небудь диска
	DirListBox (Список каталогів)	Вибір користувачем каталога на диску

	<i>FileListBox</i> (Список файлів)	Вибір користувачем файлу з каталога
	<i>Shape</i> (Фігура)	Відображення геометричних фігур у формі
	<i>Line</i> (Лінія)	Зображення графічних ліній
	<i>Image</i> (Зображення)	Виведення графічних елементів у формі (не може використовуватися як контейнер)
	<i>Data</i> (Дані)	З'єднання з існуючою базою даних
	<i>OLE</i>	Додавання в додаток функцій інших програмних засобів

Щоб ідентифікувати будь-який значок елемента на панелі інструментів Visual Basic, можна встановити на нього покажчик миші, при цьому з'явиться спливаюча підказка з назвою цього елемента.

При складанні своєї програми кожен користувач визначається, з якими конкретно об'єктами він буде працювати - з якими формами вікон, з якими пристроями (результат виводити на екран, на друк), які керуючі елементи повинні містити його вікна.

Усі вимоги до обраних об'єктів користувач фіксує у виді **властивостей** у вікні **Properties**. Кожному об'єктові відповідає свій заздалегідь заданий набір властивостей.

2.3. Властивість об'єкта

Властивість об'єкта - це якість або характеристика об'єкта, вона визначає, як об'єкт виглядає, або як він поводить. Наприклад, властивостями об'єктів є їхні імена, напис об'єкта, колір фону, символів. Список властивостей для кожного об'єкта визначений, а користувач може встановлювати конкретні значення потрібних властивостей, наприклад, колір **-блакитний**, тип шрифту - **Times New Roman**, конкретний текст у напису й т.д.

Для установки значення властивості конкретного об'єкта необхідно клацнути по цьому об'єкту, тобто зробити його активним, і вивести клацанням на передній план вікно **Properties** (рис. 2.1). У вікні відображається список властивостей активного в даний момент об'єкта й значення їх, які привласнені за замовчуванням.

Під рядком заголовка вікна розташований рядок (поле) зі списком, що випадає, який утримує перелік усіх об'єктів створеного проекту. У цьому рядку повинен бути зазначений (обраний) цікавлячий Вас об'єкт (його ім'я).

Виклик необхідної властивості для установки здійснюється по його назві *в першому стовпчику* вікна **Properties**. Щоб змінити значення властивості потрібно клацнути *в другому стовпчику* вікна на кнопці праворуч наприкінці рядка властивості у списку із припустимими значеннями, що з'явиться, вибрати клацанням необхідне або активізувати рядок і ввести дані за допомогою клавіатури.

Приклади деяких властивостей:

- **Name** - встановлює ідентифікатор (ім'я) доступу до об'єкта (текстовому полю, полю написів, полю меню, що керують кнопкам і т.д.) із програми; властивість недоступна при виконанні програми. Наприклад, Name - для форми встановлює ім'я форми, використовуване в програмі. Ім'я починається з букви, складається з букв, цифр і підкреслення, має не більш 40 символів. Спочатку задається ім'я елемента, а потім уводиться для нього код програми;

- **Alignment** - вирівнювання тексту (**0** - уліво, **1** - вправо, **2** - центрування);

- **Autosize** - автоматичне керування розміром поля напису або текстового поля (**true** - автоматична зміна відповідно до розміру вмісту, **false** - розміри незмінні й зайвий уміст відтинається);

- **BackColor, ForeColor** - установка кольору фону й переднього плану об'єкта;

- **BorderStyle** - задає тип рамки для об'єкта, причому для *form* і *textbox* тільки для читання під час виконання. Припустимі значення:

- **0**- немає контуру;

- **1**- фіксований одиночний верхній контур (рядка назви й меню);

- **2**- за замовчуванням змінюваний контур;

- **3**- фіксований подвійний.

- **Caption** - текст, відображуваний у заголовку (для форми), усередині або поруч із елементом управління;

- **FontName** - встановлює або повертає шрифт, використовуваний для відображення тексту в елементах управління, а також при виконанні операцій малювання й друку. Рекомендується використовувати шрифт **Times New Roman Cyr**. **Height, Width** - зовнішня висота й ширина об'єкта. Задаються у твіпах (1 см = 567 твіпів);

- **Text** - текст у текстовому або комбінованому полі, у полі списку. Для останнього - тільки для читання під час виконання.

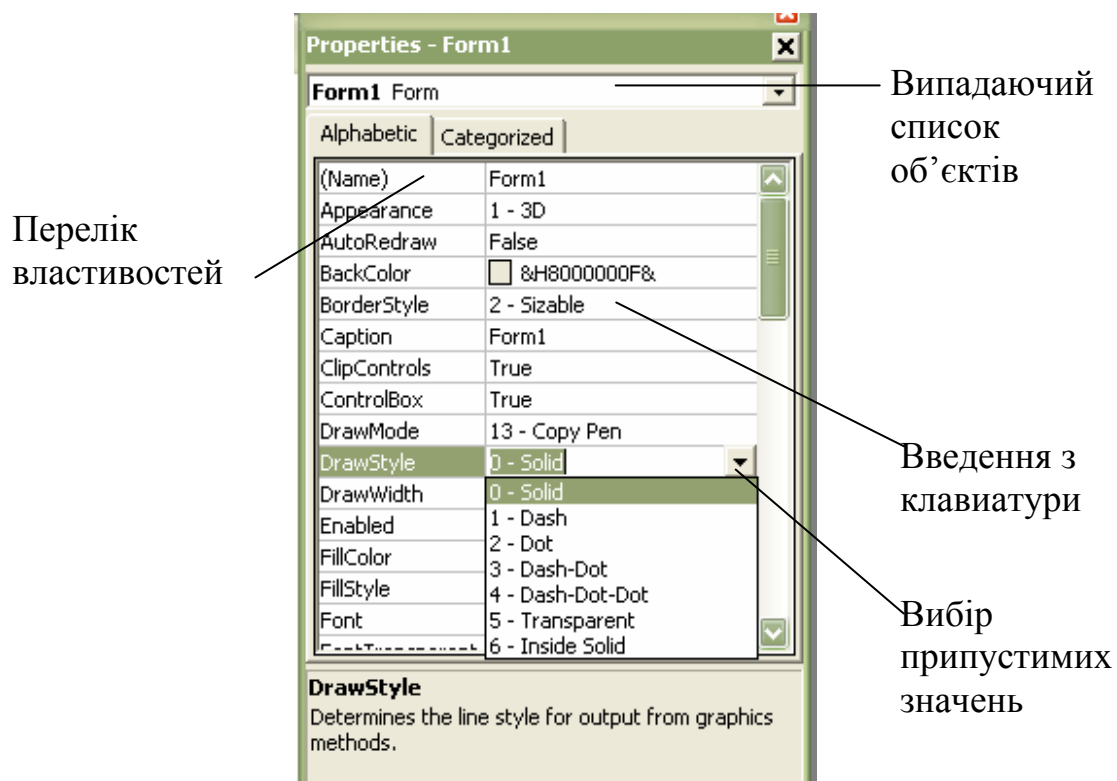


Рис. 2.1. Вікно властивостей Properties

2.4. Події

Visual Basic є мовою, орієнтованою на обробку подій. Це означає, що окремі частини програми виконуються у відповідь на певні **події**. Кожен об'єкт повинен реагувати на певні події (дії), що відбуваються:

- на екрані (клацання миші по управляючих елементах);
- на клавіатурі (натискання різних клавіш);
- у програмі під впливом конкретних даних.

Тобто програма повинна мати архітектуру, яка керована подіями.

Приклади подій:

- **Change** - відбувається при зміні стану елемента управління при наявності цілого ряду можливих подій (масиву подій);
- **Click** - відбувається при одноразовому клацанні миші над об'єктом;
- **DblClick** - відбувається при подвійному клацанні миші на об'єкті;
- **KeyPress** - відбувається при натисканні клавіші на клавіатурі;
- **Load** - відбувається при виклику додатка за допомогою оператора **Load** у програмі або в результаті неявного завантаження;
- **Unload** - відбувається при вивантаженні форми з оперативної пам'яті відповідно до дій користувача (за допомогою меню або кнопок) або оператором **Unload** із програми.

З кожною подією в Visual Basic зв'язана **процедура (підпрограма)**, що виконується при здійсненні події. У загальному виді синтаксис процедури наступний:

Sub <ім'я об'єкта>_<ім'я події> (оголошення параметрів)

.....
оператори
.....

End Sub

Для кожного об'єкта зафіксований перелік можливих процедур, задача користувача - вибрати потрібну і написати відповідний програмний код програми. Виконується це у вікні коду (рис. 2.2).

2.5. Методи

При роботі з об'єктами виникає потреба в цілому ряді стандартних операцій, зв'язаних із відповідними вікнами, кнопками, полями. Visual Basic звільняє користувача від програмування таких операцій шляхом надання набору методів.

Наприклад, можна використовувати такі методи:

- **Clear** - очищає вміст буфера обміну (**Clipboard**);
- **Cls** - очищає форму або поле ілюстрацій;
- **EndDoc** - припиняє посилку документа на принтер;
- **GetData** - повертає зображення з **Clipboard**;
- **GetText** - повертає текстовий рядок із буфера обміну;
- **Hide** - ховає форму, але не вивантажує її з оперативної пам'яті;
- **Move** - переміщає форму або об'єкт;
- **Print** - друкує текстовий рядок на об'єкті;
- **SetData** - поміщає ілюстрацію в заданому форматі в **Clipboard**;
- **SetText** - поміщає текстовий рядок у буфер обміну;
- **Show** - відображає форму на екрані.

Синтаксис запису методу:

<ім'я об'єкта> <ім'я методу> [параметри]

Приклади:

- **Form1.Hide;**
- **Form1.Show;**
- **Printer.Print "y= ";y.Text.**

Об'єкти, що мають загальні властивості й поведінку (події й методи), поєднуються в **класи**. Ім'я класу вказує на тип об'єктів, що він містить. Класи можна розглядати як основу для створення інших об'єктів цього ж типу. Тому всі об'єкти одного класу, наприклад, **TextBox**, будуть діяти однаково. За замовчуванням екземпляри того самого класу мають ім'я, що складається з імені батьківського класу й порядкового номера - **Text1**, **Text2**, **Text3**. При необхідності користувач може створювати свої класи об'єктів.

У Visual Basic також широко використовується поняття колекції об'єктів. **Колекція** - це набір об'єктів, об'єднаних загальним ім'ям, причому не обов'язково це об'єкти одного класу. Прикладом колекцій можуть служити убудовані в Visual Basic 6 колекції, у т.ч. **Forms** і **Controls**. **Forms** - це безліч

завантажених форм додатка, **Controls** - це безліч усіх елементів управління у формі. Користувач може створювати свої власні колекції, поповнювати й актуалізувати їх.

Розробка нової програми або, по термінології Visual Basic, проекту складається із двох етапів:

- **Етап візуального програмування** - розробка форми або набору форм, що будуть видаватися користувачеві під час роботи з програмою. Узгодження переліку й змісту форм із замовником може значно заощадити кошти на розробку проекту;

- **Етап програмування у вхідному коді** - розробка власної програми обробки інформації, що відповідає різним діям користувача.

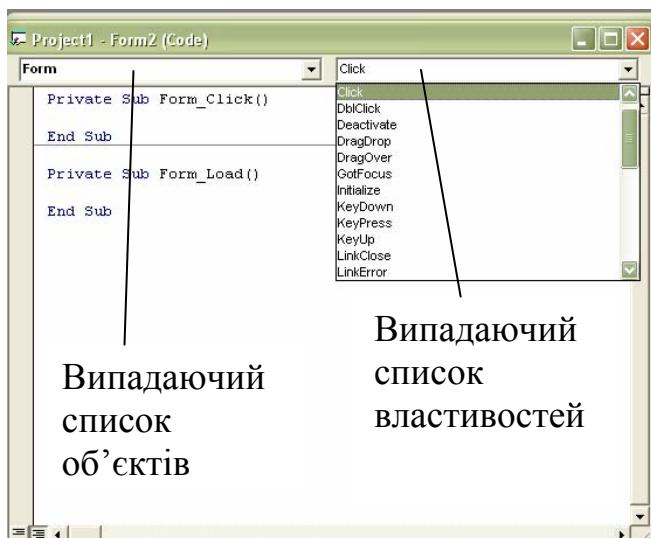


Рис. 2.2. Вікно програмного коду

3. Лабораторна робота № 1 "Визначення параметрів руху автомобіля"

Завдання: З початку руху автомобіль рухається з прискоренням. Шлях, пройдений автомобілем в метрах розраховується по формулі

$$S=1,1 \cdot t^2, \text{ де } t - \text{ час руху автомобіля, с.}$$

Швидкість автомобіля (км/ч) в кожен момент часу визначається виразом

$$V=7,92 \cdot t.$$

Спроекувати форму для розрахунку відстані, пройденої автомобілем і його швидкості в кожен момент часу.

Виконання. На екранній формі повинні (рис. 3.1) розташовуватися текстове вікно для сприйняття значень t , а також вікно для виводу результатів S та V . Щоб знати, де яке вікно знаходиться, їх необхідно підписати, створивши до них поля написів. Розрахунок повинен виконуватися при клацанні по кнопці **Розрахунок**, а по кнопці **Вихід** - здійснюється завершення роботи програми. Кнопка **"Очищення текстових вікон"** призначена для очищення текстових вікон.

Необхідні операції по прикладу приведені далі:

1. Намалюйте на папері або уявіть собі думкою вигляд такого вікна, після цього можна приступити до створення екранної форми на комп'ютері.

2. Завантажте Visual Basic.

3. У відкритому вікні Visual Basic з'явиться порожня екранна форма **Form1**. Якщо її розміри не задовольняють користувача, то варто клацнути по ній (виділити) і змінити розмір, як у звичайному малюнку.


4. Усі наступні дії по розробці форми програми представлені далі:


• дати ім'я електронній формі:

• відкрити вікно **Properties**, якщо воно відсутнє на екрані: пункт меню **View** → **Properties Windows**;

• вибрати властивість **Caption**, у правому стовпчику стерти **Form1** і написати **Розрахунок параметрів руху**;

• занести напису початкових даних і результату:

• на панелі інструментів виконати подвійне клацання по кнопці  **Frame** (Рамка). У центрі форми з'явиться рамка **Frame1** (рис. 3.1), у яку будуть поміщені початкові дані. У списку властивостей вибрати **Caption** (Назва) і привласнити назву **Початкові дані**. Аналогічно створити рамку **Frame2** з назвою **Результати обчислень**;

• на панелі інструментів виконати подвійне клацання по кнопці  **Label** - (Напис), у центрі форми з'явиться прямокутник з написом **Label1**. Перетягніть його на відведене для нього місце в створюваній формі. Ще двічі повторіть цю послідовність дій, одержавши **Label2**, **Label3**;

• активізувати клацанням миші вікно **Properties**; у ньому у верхньому списку, що випадає, вибрати **Label1**; у списку властивостей виділити **Caption**, у другому стовпчику стерти **Label1** написати: **Час руху автомобіля, t**;

• у верхньому списку вікна **Properties**, що випадає, вибрати **Label2** і у властивості **Caption** написати: **Відстань, пройдена автомобілем, S**;

- у верхньому списку вікна **Properties**, що випадає, вибрати **Label3** і у властивості **Caption** написати: **Швидкість автомобіля, V**.

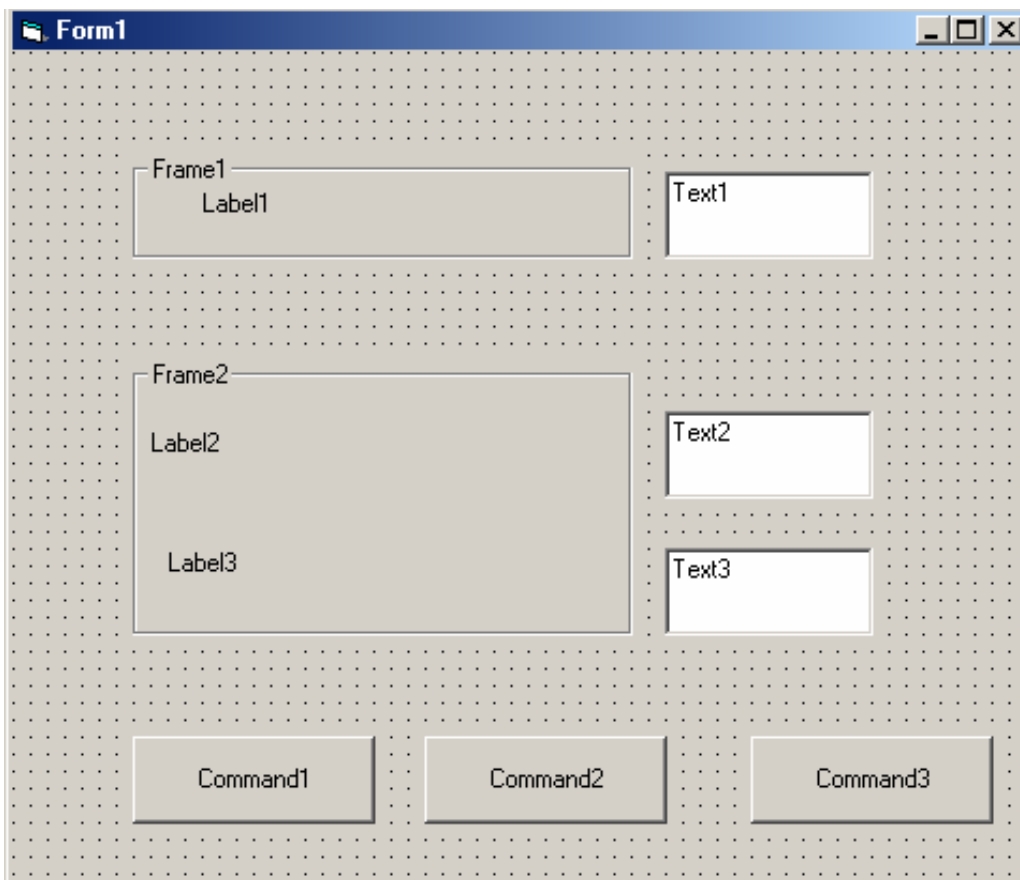




Рис. 3.1. Розташування компонентів, що управляють, на поверхні форми

- створення текстових вікон для введення і виводу даних:

- на панелі інструментів виконати подвійне клацання по кнопці  **TextBox** для створення текстового вікна **Text1**. Перетягніть його на відведене для нього місце в створюваній формі. Ще двічі повторіть цю послідовність дій, одержавши **Text2** і **Text3**;


- створення екранних кнопок:

- на панелі інструментів виконати подвійне клацання по кнопці  (**CommandButton** - командна кнопка) і перетягнути отриману кнопку на відведене для неї місце у формі, повторити зазначені дії для одержання для двох інших кнопок;

- активізувати клацанням миші вікно **Properties**; у ньому у верхньому списку, що випадає, вибрати **CommandButton1**; у списку властивостей виділити **Caption**, стерти **CommandButton1** і написати **Розрахунок**; потім виділити **Name** і замість **CommandButton1** написати **Rozraxunok**;

- у верхньому списку вікна **Properties**, що випадає, вибрати **CommandButton2**; у **Caption** написати **Очищення текстових вікон**, а в **Name** написати **Ochistka**;

- у верхньому списку вікна **Properties**, що випадає, вибрати **CommandButton3**; у **Caption** написати **Вихід**, а в **Name** написати **Vuxid**.

5. У результаті виконаних дій створена форма потрібного виду (рис. 4.4), яку необхідно зберегти - клацнути на піктограмі  стандартної панелі інструментів.

6. Двічі клацнути по кнопці **Розрахунок** і з'явиться вікно програмного коду (**Code**), у якому:

- у лівому списку, що випадає, (**Object**) - міститься перелік усіх елементів проекту;

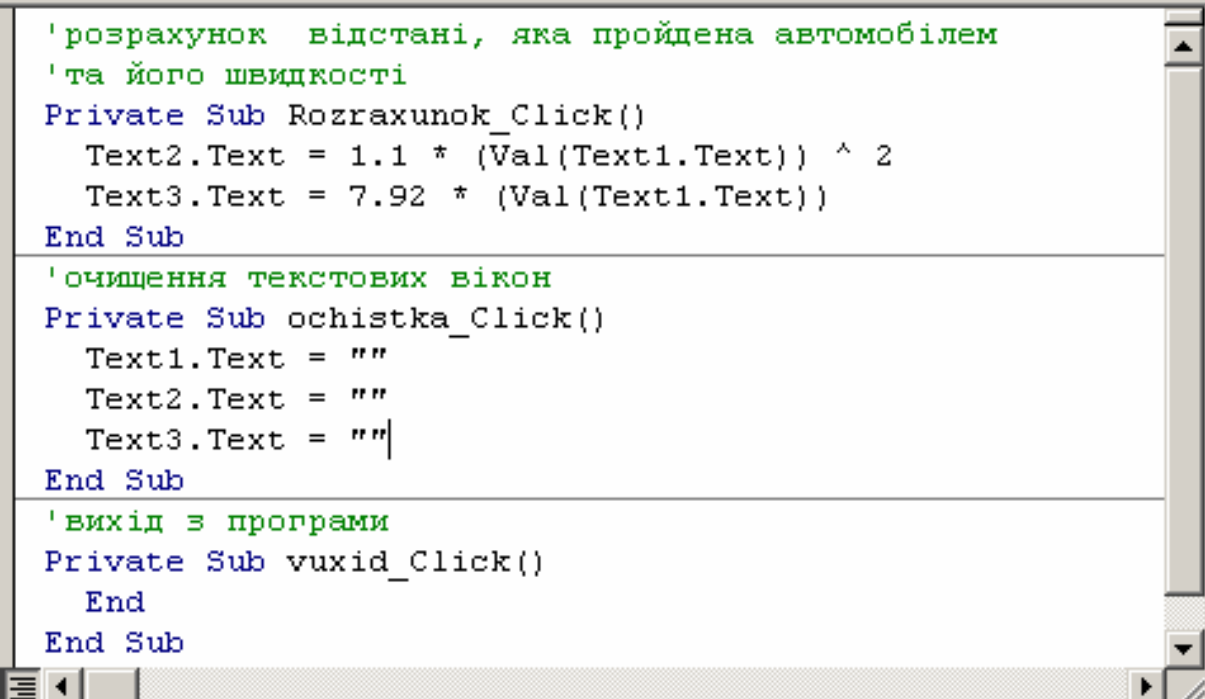
- у правому списку, що випадає, (**Procedure**) - відповідні їм процедури.

7. Вибрати:

- в лівому списку елемент **Rozрахunok**, а в правому - процедуру **Click**.

Відкриється процедура, що містить початок і кінець (рис. 3.2).

Оскільки вхідні дані й результати вводяться через текстові вікна, а в програмі йде обчислення із числами, використовується функція **Val** для перетворення текстових даних у числові;



```
'розрахунок відстані, яка пройдена автомобілем
'та його швидкості
Private Sub Rozрахunok_Click()
    Text2.Text = 1.1 * (Val(Text1.Text)) ^ 2
    Text3.Text = 7.92 * (Val(Text1.Text))
End Sub

'очищення текстових вікон
Private Sub ochistka_Click()
    Text1.Text = ""
    Text2.Text = ""
    Text3.Text = ""
End Sub

'вихід з програми
Private Sub vuxid_Click()
    End
End Sub
```

Рис. 3.2. - Вікно програмного коду проекту - "Розрахунок параметрів руху"

- у лівому списку, що випадає, викликати елемент **Ochistka**, а у правому - **Click**. Увести оператор закінчення роботи програми - **End**;

- у лівому списку, що випадає, викликати елемент **Vuxid**, а у правому - **Click**. Увести **Text1.Text = ""**

Text2.Text = ""

Text3.Text = ""

8. После преобразования и присвоения названия компонентам вид формы выглядит в соответствии с рис. 3.3;

9. Виконати написану програму:

меню **Run** → команда **Start** → ввести у вікно **Час руху автомобіля, t, сек**, - число **10**, клацнути по кнопці **Розрахунок S та V** у вікні **Відстань, яка пройдена автомобілем, S, м** повинно вийти число **110**, а у вікні **Швидкість автомобіля, V, км/год**, - число **79,2**. У випадку допущених помилок видається повідомлення й у тексті програми помилковий оператор виділяється кольором. Варто виправити помилку й знову спробувати виконати програму: меню **Run** → команда **Continue**;

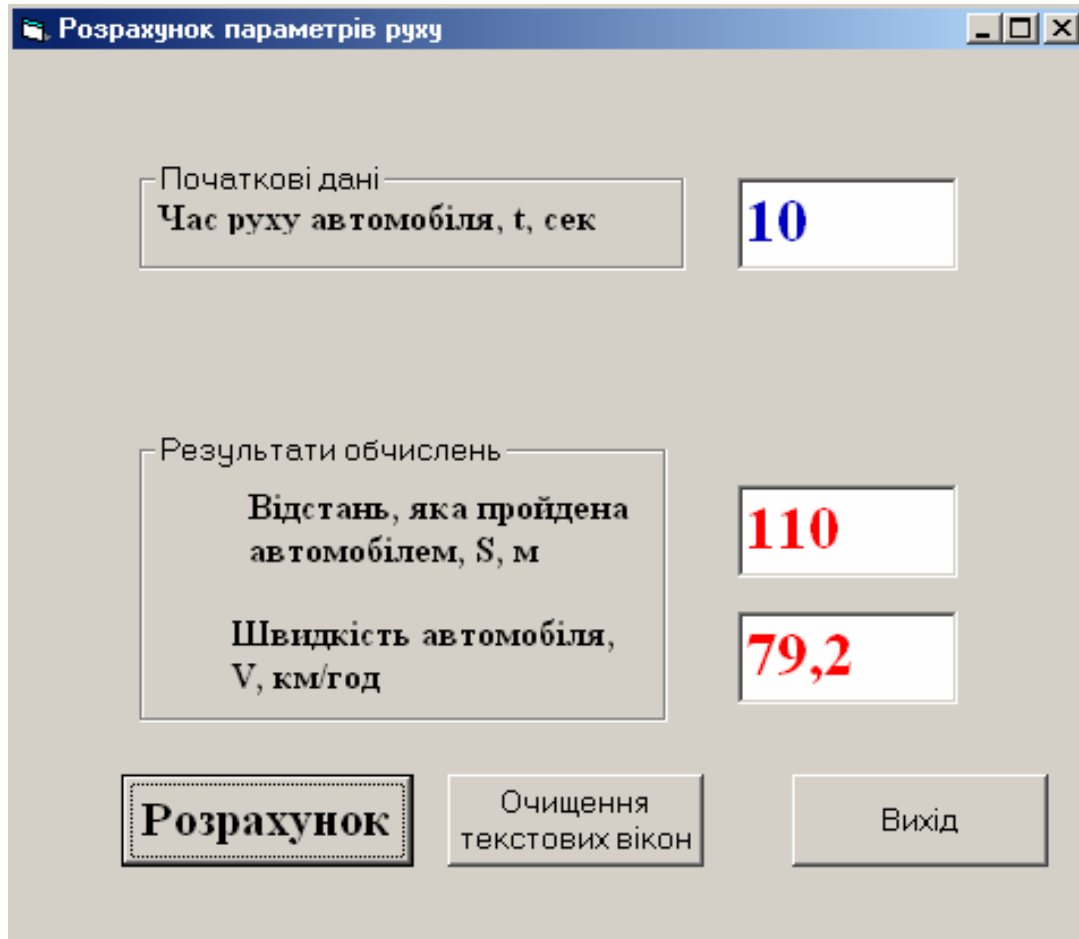


Рис. 3.3. Вікно форми проекту - "Розрахунок параметрів руху"

9. завершення роботи з програмою виконується клацанням по кнопці **Вихід**.

4. Лабораторная работа № 2

Завдання: - Розробити й налагодити програму для обчислення функції Z в розгалуженому процесі:

$$Z = \begin{cases} xy - xy^2 & \text{при } x \leq 2 \text{ і } y \geq 6 \\ \frac{(x+y)^2}{x^2+1} & \text{при } x \geq 0 \text{ і } y \geq 2 \\ x^3 + y + e^{2,5x} & \text{в інших випадках} \end{cases}$$

Виконання: Спочатку необхідно спроектувати електронну форму (див. рис. 4.1).

Занесення формули рекомендується виконати в текстовому редакторі *Word*, а потім скопіювати в керуючий елемент **OLE**.

Послідовність операцій по виконанню лабораторної роботи описані далі:

1. Створення електронної форми й завдання властивостей її елементів.

- дії аналогічні попереднім прикладам. Імена компонентів полів приведені на рис. 4.1.

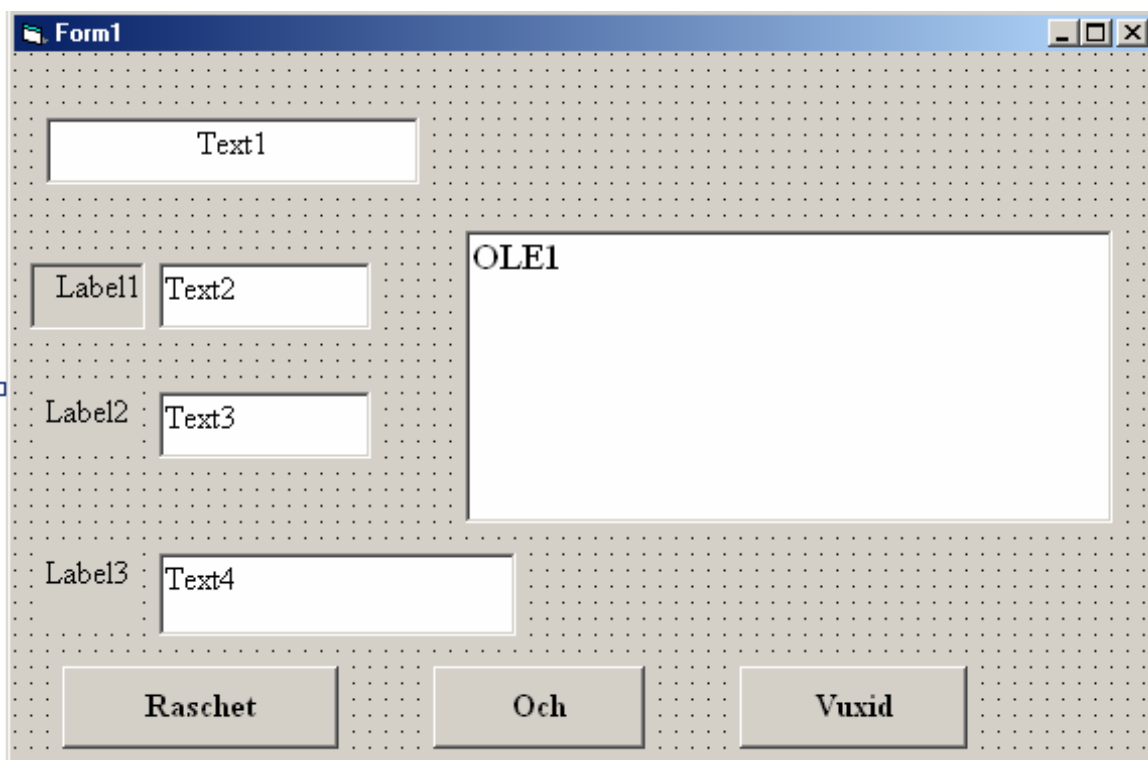


Рис. 4.1. Початкові дані для створення форми для приклада 4.5

- послідовність дій по створенню форми "Складне розгалуження" приведена в таблиці 4.1:

Таблиця 4.1.

Послідовність дій по створенню форми - "Складне розгалуження"

№ п/п	Операція	Властивість об'єкта	Значення властивості
1	Дати ім'я електронній формі.	Name	<i>Form1</i>
		Caption	<i>Складне розгалуження</i>
2	Створити текстове вікно для відображення групи і прізвища студента (інструмент TextBox)	Name	<i>Text1</i>
		Text	<i>Проект розробив студент гр. АП-08-1</i>
		Multline	<i>True</i>
3	Створити текстове вікно для введення змінної X (інструмент TextBox)	Name	<i>Text2</i>
		Text	порожньо
4	Створити текстове вікно для введення змінної Y (інструмент TextBox)	Name	<i>Text3</i>
		Text	порожньо
5	Створити текстове вікно для виводу змінної Z (інструмент TextBox)	Name	<i>Text4</i>
		Text	порожньо
6	Створити етикетку для ідентифікації змінної X (інструмент Label)	Label1	X =
7	Створити етикетку для ідентифікації змінної Y (інструмент Label)	Label2	Y =
8	Створити етикетку для ідентифікації змінної Z (інструмент Label)	Label3	Z=
9	Створити командну кнопку для виконання розрахунку функції Z (інструмент CommandButton)	Name	<i>Raschet</i>
		Caption	<i>Розрахунок</i>
10	Створити командну кнопку для очищення полів X і Y (інструмент CommandButton)	Name	<i>Och</i>
		Caption	<i>Очищення</i>

11	Створити командну кнопку для виходу з програми (інструмент CommandButton)	Name	<i>Vuxid</i>
		Caption	<i>Buxid</i>
12	Создать окно для отображения функции Z (інструмент OLE)	Name	OLE1

2. Повний програмний код приводиться далі:

```
Private Sub Raschet_Click()
'Оголошення типу даних
  Dim x, y, z As Single
'Перетворення строкових даних в числові
  x = Val(Text2.Text)
  y = Val(Text3.Text)
  z = Val(Text4.Text)
'Вибір функції
  If x <= 2 And y >= 6 Then
    z1 = x * y - x * y ^ 2
    Text4.Text = Str(Format$(z1, "0.00"))
  ElseIf x >= 0 And x >= 2 Then
    z1 = (x + y) ^ 2 / (x ^ 2 + 1)
    Text4.Text = Str(Format$(z1, "0.00"))
  Else
    z1 = x ^ 3 + y + Exp(2.5 * x)
    Text4.Text = Str(Format$(z1, "0.00"))
  End If
End Sub

```

```
'Процедура виходу з програми
Private Sub Vuxid_Click()
  End
End Sub

```

```
'Очищення текстових вікон
Private Sub Och_Click()
  Text2.Text = ""
  Text3.Text = ""
  Text4.Text = ""
End Sub

```

3. Остаточний вигляд екранної форми приведений на рис. 4.2.

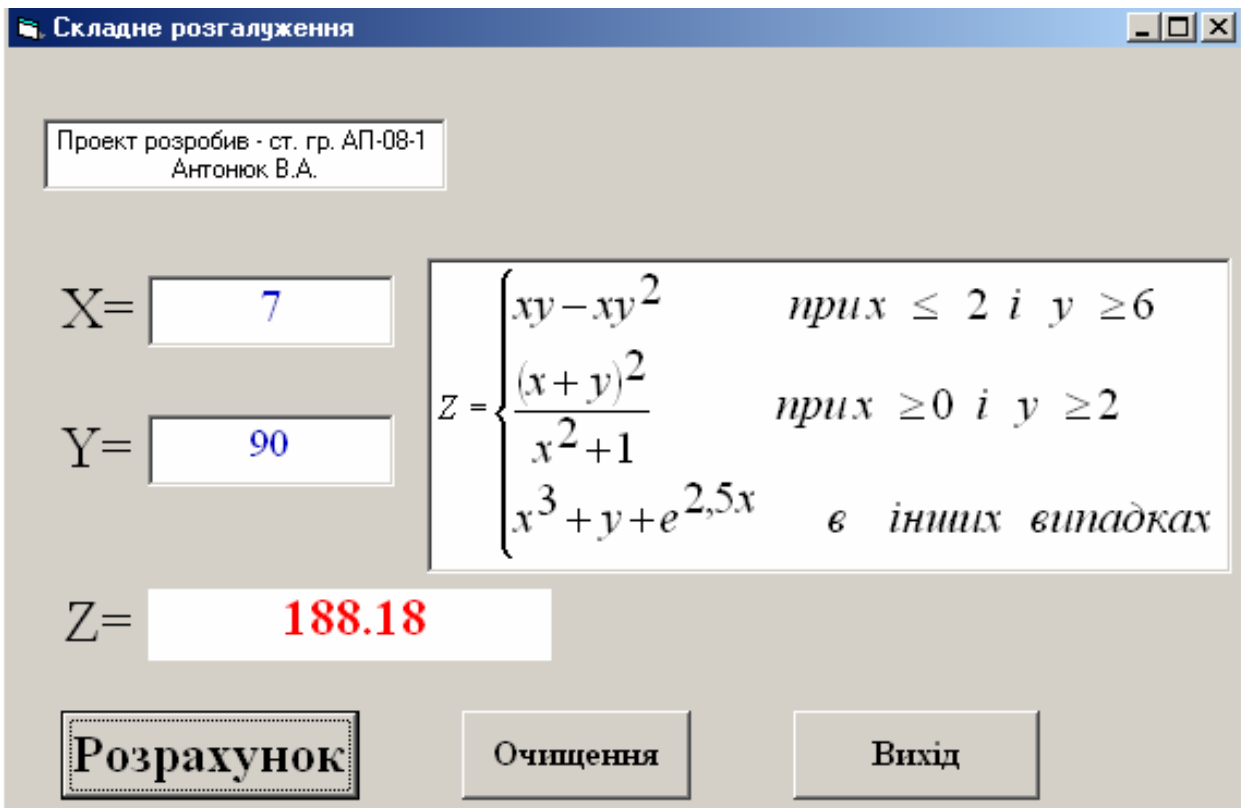


Рис. 4.2. Вигляд екранної форми для прикладу 4.3 - "Складне розгалуження"

4. Робота з програмою. Для роботи з програмою необхідно спочатку ввести початкові дані: змінну **X** у вікно з підказкою "X=", а змінну **Y** у вікно з підказкою "Y=", а потім клацнути лівою клавiшею миші на кнопці "Розрахунок". Результат розрахунку функції **Z** виводиться у вікно з підказкою "Z=".

5. Лабораторна робота № 3 "Вибір функції"

Завдання: обчислити значення функції по одній із трьох формул:
 $y = e^x + x^2$, $y = x^4 + x^2 + 1$ або $y = x^2 - 1$ для $0 < x \leq 100$

Для вибору формул при розрахунку використовувати перемикачі. Значення перемінної x задавати за допомогою лінійки прокручування.

Виконання:

1) Створити екранну форму (рис. 5.1)

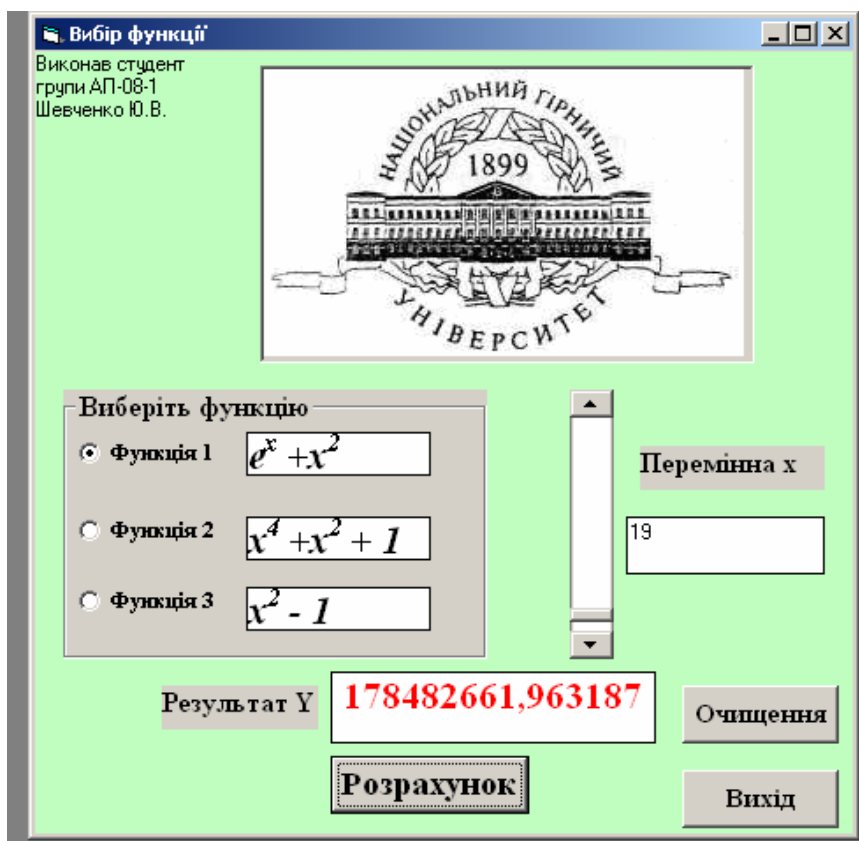


Рис. 5.1. Вигляд екранної форми для лабораторної роботи- "Вибір функції"

Послідовність дій по створенню форми "Вибір функції" приведена в таблиці 5.1.

Таблиця 5.1

Послідовність дій по створенню форми - "Вибір функції"

№ п/п	Операція	Властивість об'єкта	Значення властивості
1	Дати ім'я електронній формі.	Caption	Вибір функції
2	Створити лінійку прокручування для вибору значень перемінної X (інструмент HScrollBar)	Name	LX
3	Створити рамку для групи перемикачів (інструмент Frame)	Caption	Виберіть функцію

4	Створити перемикач для вибору функції 1 (інструмент OptionButton)	Name	F1
5	Створити перемикач для вибору функції 2 (інструмент OptionButton)	Name	F2
6	Створити перемикач для вибору функції 3 (інструмент OptionButton)	Name	F3
7	Створити у формі для ілюстрації розрахункову формулу $y = x / 6$ (інструмент OLE)	Name	OLE1
8	Аналогічно п.7 дві інші формули	Name	OLE2
9	Створити етикетку для ідентифікації поля зі значенням перемінної x (інструмент Label)	Name	Label1
10	Створити поле для перегляду значення перемінної x (інструмент Text)	Name	X
11	Створити етикетку для ідентифікації поля результату розрахунків y (інструмент Label)	Name	Label2
12	Створити поле для перегляду значення результату y (інструмент Text)	Name	y
13	Створити командну кнопку для виконання розрахунку по обраній формулі (інструмент CommandButton)	Name	Raschet
		Caption	Розрахунок
14	Створити командну кнопку для очищення полів x і y (інструмент CommandButton)	Name	Och
		Caption	Очищення
15	Створити командну кнопку для виходу з програми (інструмент CommandButton)	Name	Vuxid
		Caption	Вихід

2) Використовуючи оператора **Print**, помістити на поверхню форми зверху з лівого боку прізвище студента, ініціали і назву групи, в якій він вчиться. Програмний код для виконання цієї дії наступний:

```
Private Sub Form_Load()
    Print "Віконав студент"
    Print "групи АП-08-1"
    Print "Шевченка Ю.В."
```

End Sub

У верхній частині форми в центрі помістити емблему Національного гірничого університету. Растрове зображення емблеми вставляється у вікно, що

створюється інструментом  **PictureBox**.

3) Створити програмний код.

- Подвійним клацанням лівої кнопки мишки по лінійці прокручування в формі проекту відкрити вікно **Code**.

Для об'єкта **Line** вибрати подію **Change** і дописати процедуру, виконувану, якщо вже відбудеться подія - переміщення бігунка на лінійці:

```
Private Sub LX_Change()  
    x.Text = LX.Value  
End Sub
```

● Для об'єкта **Line** вибрати подію **Scroll** і дописати зміст процедури, виконуваної під час переміщення бігунка по лінійці:

```
Private Sub LX_Scroll()  
    LX_Change  
End Sub
```

● Вибрати в лівому списку об'єкт **Розрахунок**, а в правому - подію **Click**. Описати послідовність дій, виконуваних у програмі при клацанні мишкою по кнопці "**Розрахунок**":

```
Private Sub Raschet_Click()  
    If F1.Value = True Then  
        y.Text = Exp(Val(x.Text)) + (Val(x.Text)) ^ 2  
    ElseIf F2.Value = True Then  
        y.Text = (Val(x.Text)) ^ 4 + (Val(x.Text)) ^ 2 + 1  
    Else  
        y.Text = (Val(x.Text)) ^ 2 - 1  
    End If  
End Sub
```

● Вибрати в лівому списку об'єкт **Очищення**, а в правому - подію **Click**. Описати послідовність дій, виконуваних у програмі при клацанні мишкою по кнопці "**Очищення**":

```
Private Sub Och_Click()  
    x.Text = ""  
    y.Text = ""
```

End Sub

• Вибрати в лівому списку об'єкт **Вихід**, а в правому - подію **Click**.
Описати послідовність дій, виконуваних у програмі при клацанні мишкою по кнопці "**Вихід**"

Private Sub vuxid_Click()

End

End Sub

У лабораторній роботі необхідно було для виконання розрахунків вибрати одну із трьох формул, якщо ж потрібно робити вибір типу "**Так/Ні**", то варто використовувати прапорець. Наявність "галочки" у квадратику відповідає "**Так**" (властивість **Value - 1 Checked**), відсутність - "**Ні**" (властивість **Value - 0 Unchecked**).

```
Private Sub Form_Load() 'сведения о студенте
Print "Виконав студент"
Print "групи АП-08-1"
Print "Шевченко Ю.В."
End Sub
```

```
Private Sub LX_Change() 'процедура свойств линейки прокрутки
x.Text = LX.Value
End Sub
```

```
Private Sub LX_Scroll() 'процедура выполнения прокрутки
LX_Change
End Sub
```

```
Private Sub Och_Click() ' процедура очистки текстовых окон
x.Text = ""
y.Text = ""
End Sub
```

```
Private Sub Raschet_Click() 'процедура расчета
If F1.Value = True Then
y.Text = Exp(Val(x.Text)) + (Val(x.Text)) ^ 2
ElseIf F2.Value = True Then
y.Text = (Val(x.Text)) ^ 4 + (Val(x.Text)) ^ 2 + 1
Else
y.Text = (Val(x.Text)) ^ 2 - 1
End If
End Sub
```

```
Private Sub vuxid_Click() 'процедура выхода из программы
End
End Sub
```

|

6. Лабораторна робота № 4 "Розрахунок гальмівного шляху автомобіля"

Завдання: Автомобіль рухається по дорозі із швидкістю V км/год. Через певні обставини водій вимушений виконати екстрене гальмування. Розрахувати гальмівний шлях автомобіля залежно від стану дорожнього покриття, що обумовлене в даному завданні коефіцієнтом зчеплення f коліс автомобіля з дорогою. Для введення швидкості руху автомобіля передбачити в програмі лінійку прокручування, а для вибору коефіцієнта зчеплення f використовувати перемикачі.

Виконання:

1) Створити екранну форму згідно рис. 6.1.

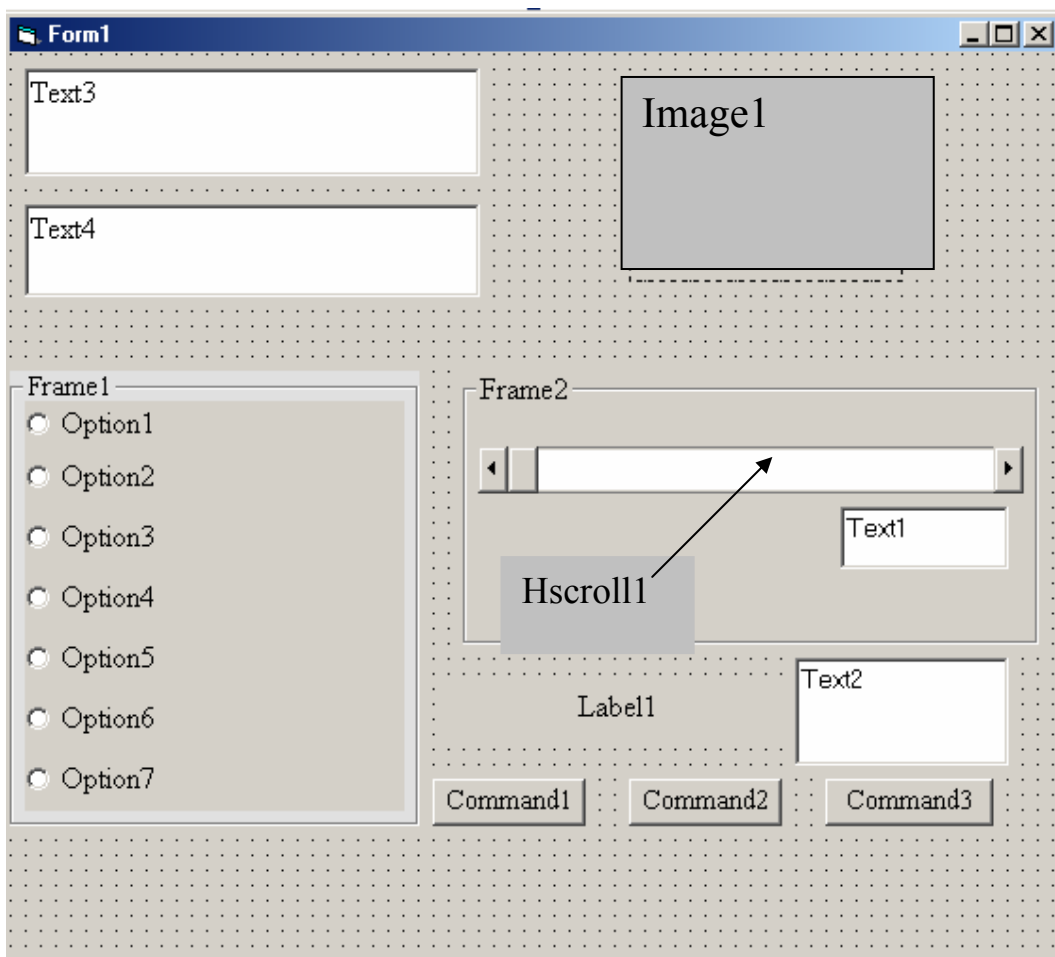


Рис. 6.1. Початкові дані для створення форми лабораторної роботи № 5

Послідовність дій по створенню форми "Вибір функції" приведена в таблиці 6.1

Послідовність дій по створенню форми – «Вибір функції»

№ п/п	Операція	Властивість об'єкта	Значення властивості
1	Дати ім'я електронній формі.	Name	<i>Form1</i>
		Caption	<i>Гальм_шлях</i>
2	Створити текстове вікно для відображення значень гальмівного шляху	Name	<i>Text3</i>
		Text	<i>Розрахунок гальмівного шляху автомобіля</i>
		Multline	<i>True</i>
3	Створити текстове вікно для відображення значень гальмівного шляху	Name	<i>Text4</i>
		Text	Прізвище студента, група
		Multline	<i>True</i>
4	Створити рамку для групи перемикачів (інструмент Frame)	Name	<i>Frame1</i>
		Caption	<i>Вибір коефіцієнта зчеплення f</i>
5	Створити перемикач для вибору коефіцієнту $f=0,1$ (інструмент OptionButton)	Name	<i>Option1</i>
		Caption	<i>$f=0,1$</i>
6	Створити перемикач для вибору коефіцієнту $f=0,2$ (інструмент OptionButton)	Name	<i>Option2</i>
		Caption	<i>$f=0,2$</i>
7	Створити перемикач для вибору коефіцієнту $f=0,3$ (інструмент OptionButton)	Name	<i>Option3</i>
		Caption	<i>$f=0,3$</i>
8	Створити перемикач для вибору коефіцієнту $f=0,4$ (інструмент OptionButton)	Name	<i>Option4</i>
		Caption	<i>$f=0,4$</i>
9	Створити перемикач для вибору коефіцієнту $f=0,5$ (інструмент OptionButton)	Name	<i>Option5</i>
		Caption	<i>$f=0,5$</i>
10	Створити перемикач для вибору коефіцієнту $f=0,6$ (інструмент OptionButton)	Name	<i>Option6</i>
		Caption	<i>$f=0,6$</i>
11	Створити перемикач для вибору коефіцієнту $f=0,7$ (інструмент OptionButton)	Name	<i>Option7</i>
		Caption	<i>$f=0,7$</i>

12	Создать рамку для компонентов ввода скорости движения автомобиля (инструмент Frame)	Name	<i>Frame2</i>
		Caption	<i>Введення швидкості руху автомобіля</i>
	Створити лінійку прокручування для вибору значень перемінної V (інструмент HScrollBar)	Name	<i>HScroll1</i>
		Large Change	<i>1</i>
		Max	<i>250</i>
		Min	<i>0</i>
	Створити текстове вікно для відображення значень швидкості автомобіля, що вводяться	Name	<i>Text1</i>
		Text	порожньо
		Font	Шрифт- <i>Times New Roman</i> Начертание- <i>Жирный курсив</i> Размер- <i>12</i>
	Створити текстове вікно для відображення значень гальмівного шляху	Name	<i>Text2</i>
		Text	порожньо
		Font	Шрифт- <i>Times New Roman</i> Начертание- <i>Жирный курсив</i> Размер- <i>16</i>
	Створити етикетку для ідентифікації поля зі гальмівного шляху (інструмент Label)	Label1	<i>Гальмівний шлях =</i>
	Створити командну кнопку для виконання розрахунку гальмівного шляху(інструмент CommandButton)	Name	<i>Command1</i>
		Caption	<i>Розрахунок</i>
	Створити командну кнопку для очищення полів x і y (інструмент CommandButton)	Name	<i>Command2</i>
		Caption	<i>Очищення</i>
	Створити командну кнопку для виходу з програми (інструмент CommandButton)	Name	<i>Command3</i>
		Caption	<i>Вихід</i>

2) У верхній частині форми з правою стороною помістити малюнок автомобіля з колекції кліпів текстового процесора Microsoft Word **Вставка**→**Рисунок**→**Картинки**. Зображення вставляється у вікно, що

створюється компонентом  **Image**

3) Створити програмний код. Повний програмний код приводиться далі:

```
' створення лінійки прокручування для введення швидкості
'автомобіля (використання компоненту HScrollBar)
Private Sub HScroll11_Change()
    Text1.Text = HScroll11.Value
End Sub
-----
Private Sub HScroll11_Scroll()
    HScroll11_Change
End Sub
-----
Private Sub Command2_Click() 'процедура очищення текстових вікон
    Text1.Text = ""
    Text2.Text = ""
End Sub
-----
' процедура обчислення гальмівного шляху з використанням
' компоненту CommandButton
Private Sub Command1_Click()
If Option1.Value = True Then
    Text2.Text = (Val(Text1.Text)) ^ 2 * 0.0039327 / 0.1
ElseIf Option2.Value = True Then
    Text2.Text = (Val(Text1.Text)) ^ 2 * 0.0039327 / 0.2
ElseIf Option3.Value = True Then
    Text2.Text = (Val(Text1.Text)) ^ 2 * 0.0039327 / 0.3
ElseIf Option4.Value = True Then
    Text2.Text = (Val(Text1.Text)) ^ 2 * 0.0039327 / 0.4
ElseIf Option5.Value = True Then
    Text2.Text = (Val(Text1.Text)) ^ 2 * 0.0039327 / 0.5
ElseIf Option6.Value = True Then
    Text2.Text = (Val(Text1.Text)) ^ 2 * 0.0039327 / 0.6
ElseIf Option7.Value = True Then
    Text2.Text = (Val(Text1.Text)) ^ 2 * 0.0039327 / 0.7
End If
End Sub
-----

Private Sub command3_Click() ' процедура виходу з програми
    End
End Sub
```

4) Остаточно вигляд форми для розрахунку гальмівного шляху приведений на рис. 6.2

Form1

Розрахунок гальмівного шляху автомобіля

Виконав студент гр. АП-08-1
Петренко Ю.А.

Вибір коефіцієнта зчеплення f

- $f = 0,1$
- $f = 0,2$
- $f = 0,3$
- $f = 0,4$
- $f = 0,5$
- $f = 0,6$
- $f = 0,7$

Введення швидкості руху автомобіля, км/год

100

Гальмівний шлях, м 78,654

Розрахунок Очищення Вихід

Рис. 6.2. Остаточний вид форми прикладної програми

5) Для розрахунку гальмівного шляху необхідно за допомогою лінійки прокручування ввести швидкість руху автомобіля (у даному прикладі 100 км./год.) і перемикачем вибрати коефіцієнт зчеплення f коліс автомобіля з дорожнім покриттям ($f=0,5$). Натиснення на кнопку **Розрахунок** приведе до появи в текстовому вікні значення гальмівного шляху (78,654 м).

7. Лабораторна робота № 5

Завдання: Створити форму (рис. 7.1) для розв'язування задачі табулювання функції. Побудувати у ній головне меню з командами:

- закінчити роботу програми,
- табулювати функцію,
- очистити поля виведення результатів.

Результати табулювання вивести у багаторядкове поле редагування (об'єкт типу **Text**).


Об'єкт **CheckBox** призначений у разі потреби виведення результатів табулювання похідній заданої функції.

Виконання.

1. Завантажте середовище Visual Basic. Вигляд форми за умовчанням приведений на рис. 7.2.

2. Змініть заголовок (**Caption**) форми з "**Form1**" на "**Табулювання функції**" (без лапок) і збільшіть розміри форми, а також ім'я форми з "**Form1**" на **Таблр**.

3. Збережіть виконану на даний момент форму у своїй власній папці (**File** → **Save Project**).


4. Вставте у форму поле редагування об'єкт  **TextBox (Text1)**. Збільшіть розміри поля. Властивість **ScrollBars** (наявність смуг прокручування) цього об'єкта задайте **Both** (будуть обидві смуги — вертикальна і горизонтальна). Властивість **MultiLine** задайте **True** - дозвіл об'єкту працювати більш ніж з одним рядком.

5. Розташуйте у формі поля редагування окна **Text2**, **Text3**, **Text4**, і відповідні їм текстові поля "**Початкове значення X**", "**Кінцеве значення X**", "**Крок зміни X**", а також поле для вигляду даної функції $Y = \text{Exp}(-X) + X^2 + X + 1$ (**Label1**). Розмір, стиль і колір шрифтів виберіть на власний розсуд так, щоб форма виглядала як найкраще. Текстове поле **Text5** призначене для виводу інформації про навчальний заклад, факультет, групу і прізвище студента, що виконує завдання.

6. Задайте початкові значення для полів редагування лівої і правої меж аргумента функції та для кроку зміни цього аргумента, наприклад, такі, як на рис. 2. Для цього змініть властивість **Text** цих об'єктів.

7. Вирівняйте вставлені поля редагування до лівого краю першого об'єкта та надайте їм однакові розміри.

8. Вставте у форму прапорцець.

Для цього використайте компоненту  **CheckBox** із палітри компонентів.

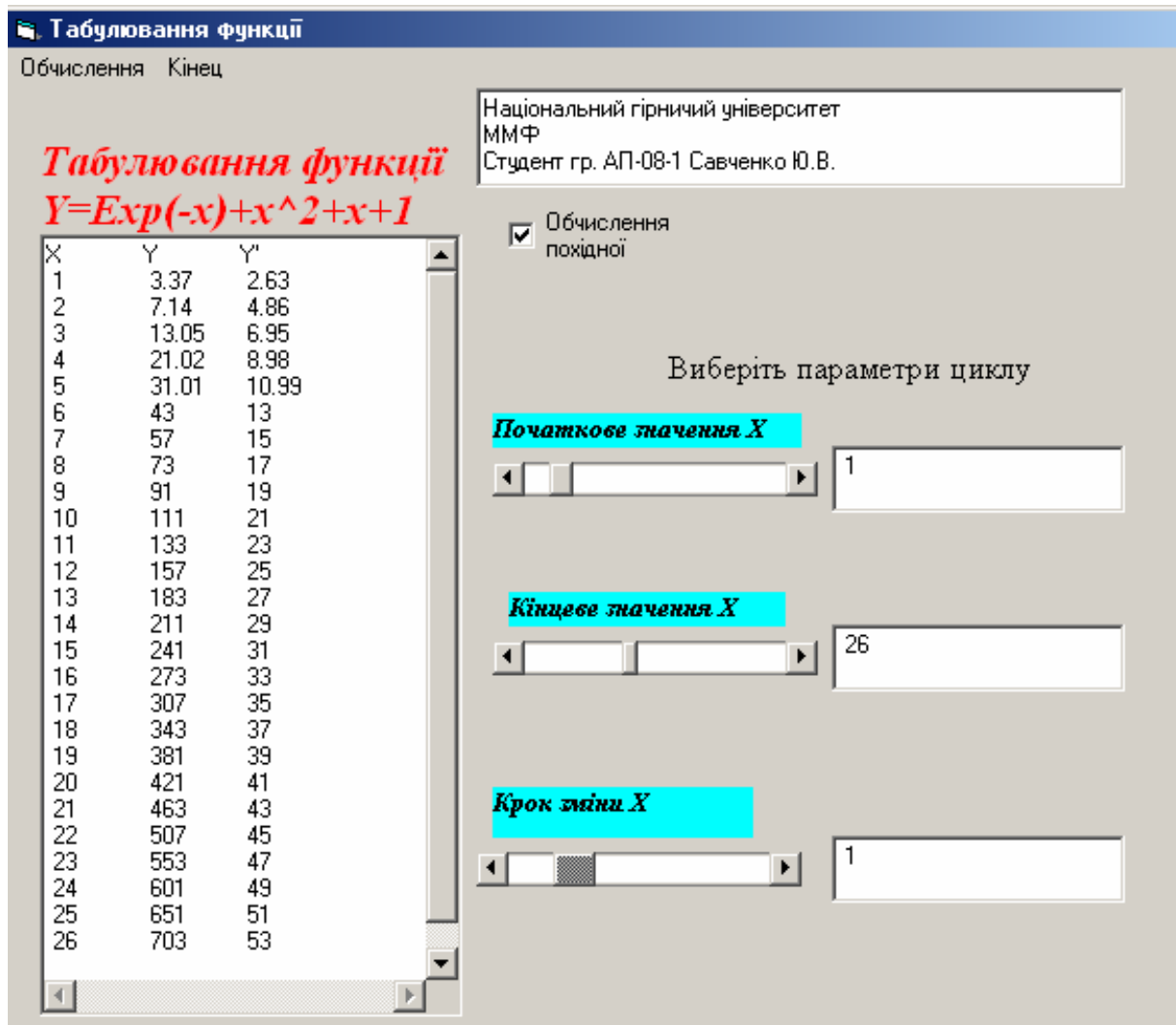


Рис. 7.1. Вигляд форми для обчислення табуляції

9. Введіть назви команд головного меню форми (рис. 7.3).

Команди головного меню, як і інші компоненти Visual Basic, є об'єктами. Для створення команд виберіть елемент головного меню **Tools** → **Menu Editor**. У вікні, що з'явиться, введіть назви команд меню (властивість **Caption**) і їхні імена (властивість **Name**), щоразу натискаючи на кнопку "Next" (табл. 7.1).

Таблиця 7.1

Назва пунктів головного меню і відповідні ним імена

Caption	Name
Обчислення	mnuCalc
Таблювання	mnuTablr
Очистити	mnuClear
Кінець	mnuFinish
Вихід	mnuEnd

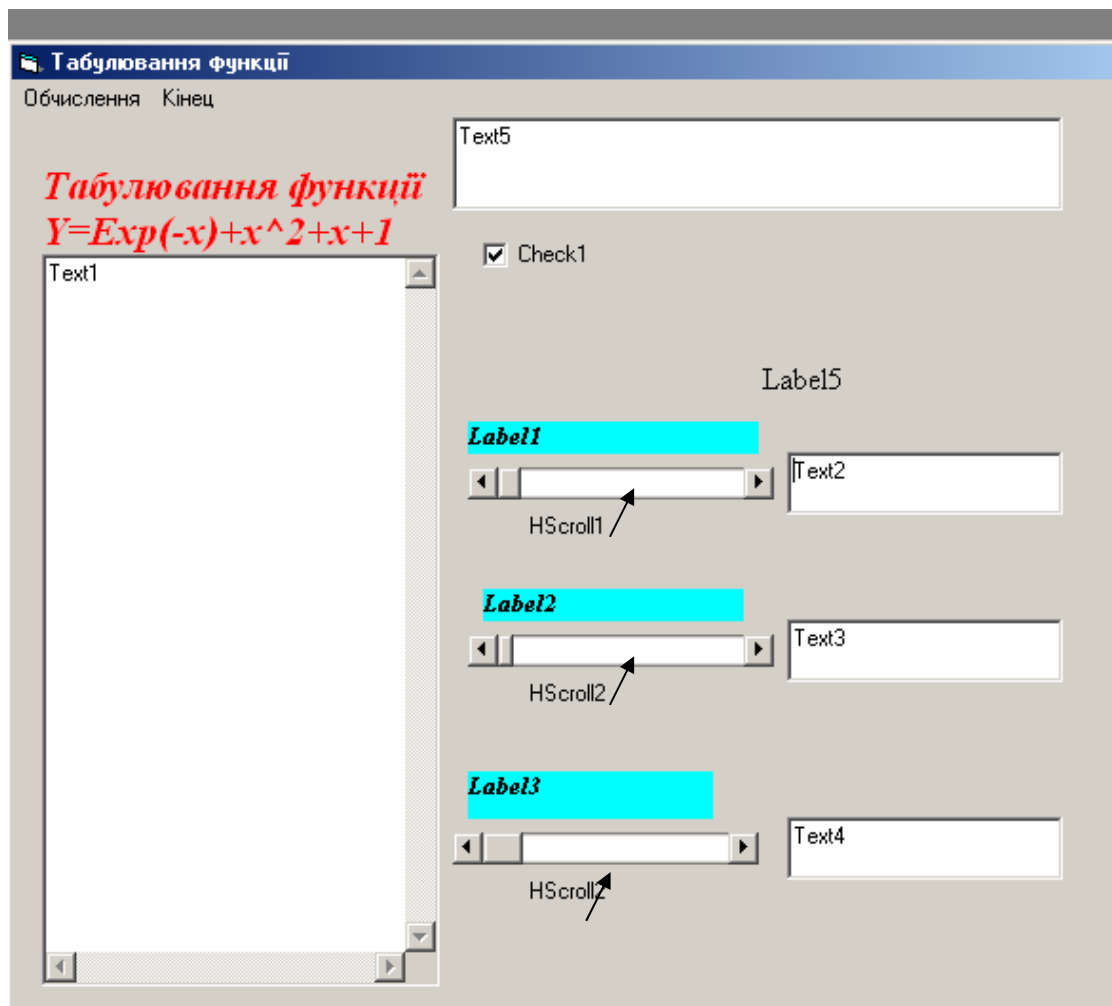






Рис. 7.2. Початкові дані для розробки форми і імена об'єктів

Задайте ієрархію команд за допомогою стрілок:  (підпорядкувати) та  (вивести із підпорядкування). Змінити послідовність команд можна за допомогою стрілок:  та .

10. Запрограмуйте команду "Очистити" головного меню. Клацніть на команді головного меню форми "Очистити", не запускаючи програму на виконання. З'явиться заготовка процедури реакції на подію виклику цієї команди. У ній запишіть команду присвоєння порожнього рядка для очистки поля редагування **Text1**, **Text2**, **Text3**, **Text4**:

Private Sub mnuClear_Click()

Text1.Text = ""

Text2.Text = ""

Text3.Text = ""

```
Text4.Text = ""  
End Sub
```

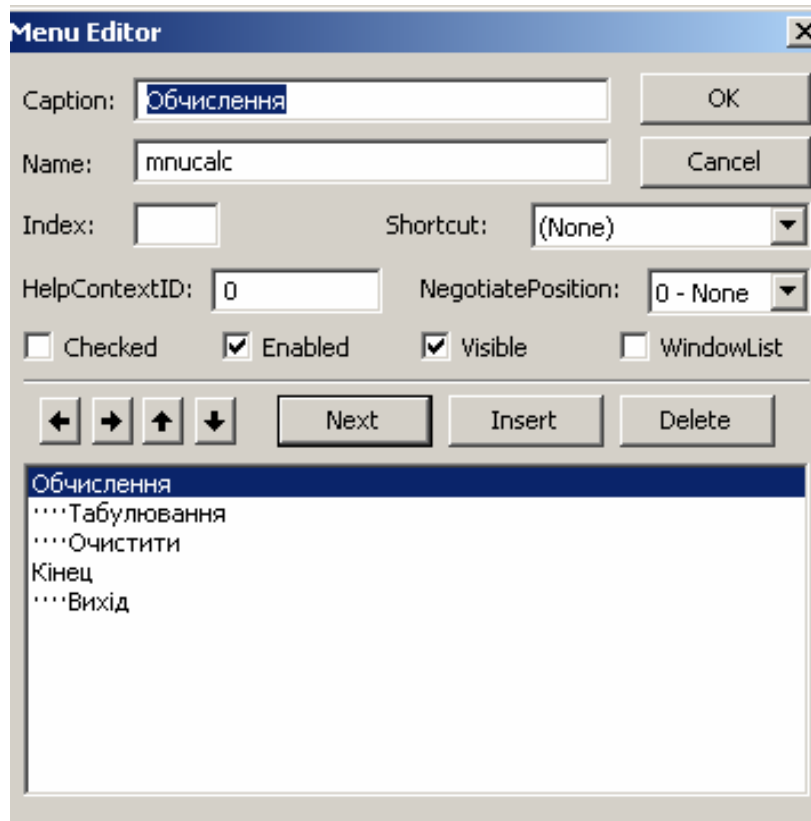


Рис. 7.3. Створення головного меню в середовищі редактора меню **Menu Editor**

11. Запрограмуйте команду "**Кінець**" головного меню, скориставшись стандартною процедурою **End**:

```
Private Sub mnuEnd_Click()  
End  
End Sub
```

12. Запрограмуйте команду "**Таблювання**".
Виконайте команду "**Таблювання**" з головного меню форми, клацнувши на ній один раз. З'явиться заготовка процедури, яку необхідно заповнити.

Весь програмний код приведений нижче:

```

Private Sub mnuclear_Click()
'очищення текстових вікон виведення результатів
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
End Sub


---


'створення смуги прокручування початкового значення X
Private Sub HScroll11_Scroll11()
HScroll11_Change
End Sub


---


Private Sub HScroll11_Change()
lb1 = Str(HScroll11)
'виведення початкового значення X у вікні Text3
'за допомогою смуги прокручування
Text2.Text = lb1
End Sub


---


'створення смуги прокручування кінцевого значення X
Private Sub HScroll12_Scroll12()
HScroll12_Change
End Sub


---


Private Sub HScroll12_Change()
lb2 = Str(HScroll12)
'виведення кінцевого значення X у вікні Text3
'за допомогою смуги прокручування
Text3.Text = lb2
End Sub


---


'створення смуги прокручування кроку зміни X
Private Sub HScroll13_Scroll13()
HScroll13_Change
End Sub


---


Private Sub HScroll13_Change()
lb3 = Str(HScroll13)
'виведення кроку зміни X у вікні Text3
'за допомогою смуги прокручування
Text4.Text = lb3
End Sub


---



```

```

Private Sub mnuend_Click() 'вихід з програми
    End
End Sub

```

```

Private Sub mnutablr_Click()
    Dim x, y As Double 'оголошення типу даних Double
    Dim NewLine, Space As String 'оголошення типу даних String
    NewLine = Chr(13) + Chr(10) 'Символ "Enter"
    Space = Chr(9) 'Символ "Tab"
    If Check1.Value = Checked Then
        'побудова шапки таблиці обчислення функції Y і її похідної Y'
        Text1.Text = "X" + Space + "Y" + Space + "Y'" + NewLine
    Else
        'побудова шапки таблиці обчислення функції Y
        Text1.Text = "X" + Space + "Y" + NewLine
    End If
    For x = Val(Text2.Text) To Val(Text3.Text) Step Val(Text4.Text)
        y = Exp(-x) + x ^ 2 + x + 1 'обчислення функції Y
        Y1 = -Exp(-x) + 2 * x + 1 'обчислення похідної Y'
        If Check1.Value = Checked Then
            'вивід в текстове вікно Text1
            'результатів обчислення функції Y та похідної Y'
            Text1.Text = Text1.Text + Str(Format$(x, "0.00")) + Space + _
                Str(Format$(y, "0.00")) + Space + Str(Format$(Y1, "0.00")) + NewLine
        Else
            'вивід в текстове вікно Text1 результатів обчислення функції Y
            Text1.Text = Text1.Text + Str(x) + Space + _
                Str(Format$(y, "0.00")) + NewLine
        End If
    Next x
End Sub

```

8. Лабораторна робота № 6 "Складний цикл"

Завдання: Обчислити значення функції

$$Y = \begin{cases} Y = \sum_{n=1}^{10} x^n / n, & \text{якщо } X < 0,55 \\ Y = \frac{X^2}{3} \prod_{k=1}^{15} (1 + x/k), & \text{якщо } X \geq 0,55 \end{cases}$$

где $x=0,1 (0,05) 0,9$

Виконання: У лабораторній роботі вирішується завдання табулювання функції. Алгоритм обчислення функції приведений на рис. 8.1.

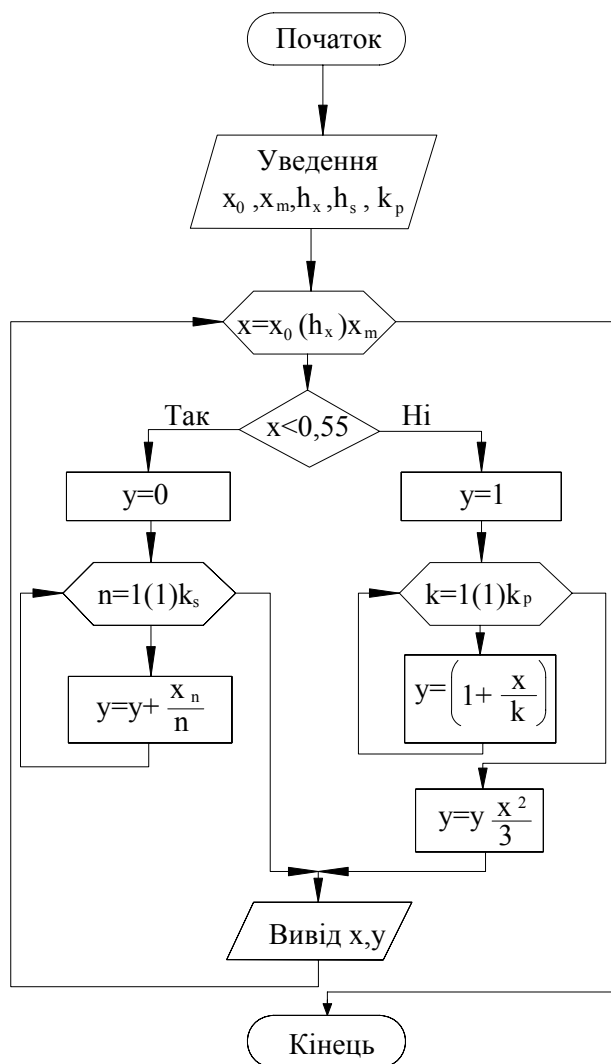


Рис. 8.1. Блок-схема алгоритму обчислення функції $Y = Y(x)$

$y = y(x)$. На першому етапі показана структура циклу із заголовком, в тілі якого міститься блок обчислення функції $y = y(x)$ і виведення значень x і y (рис. 8.1). Обчислювана функція є складною і визначається по-різному залежно від умови $x < 0,55$. В результаті на другому етапі деталізації блок обчислення функції розкритий у вигляді простого розгалуження. На третьому етапі деталізації розкриваються блоки обчислення функцій $y = y_1(x)$ і $y = y_2(x)$ у вигляді простих алгоритмів обчислення суми і твору.

1) **Створення початкової екранної форми проекту**. Початкова екранна форма створюється згідно рис. 8.2. На поверхні форми приведені імена і назви компонентів, прийняті за умовчанням. Надалі імена кожного з компонентів не змінний, а їх назви приймаємо відповідно до виконуваних і необхідних дій.

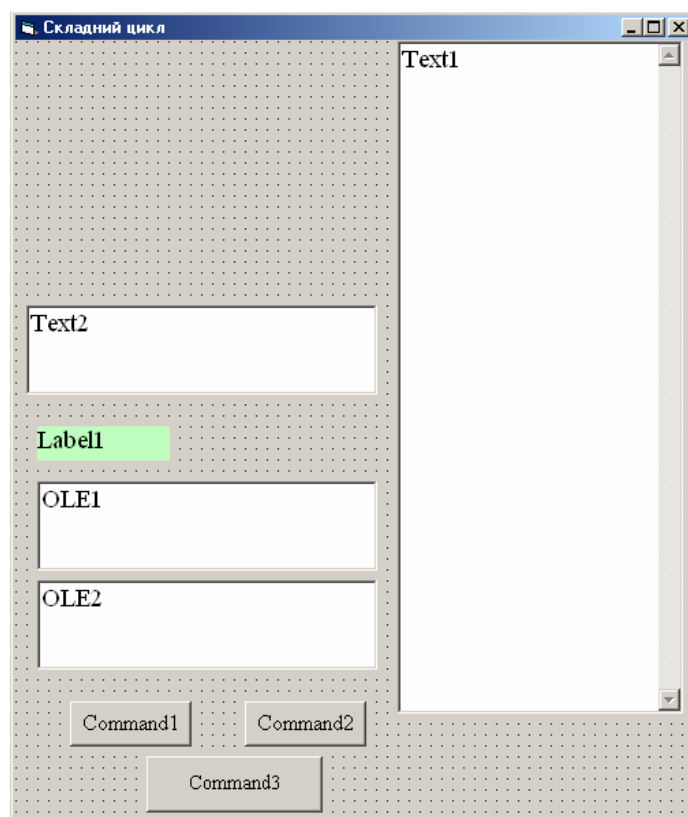


Рис. 8.2. Початкові дані для створення форми "Складний цикл"

2) **Послідовність дій по створенню форми "Складний цикл"**. Послідовність приведена в таблиці 1

Таблиця 8.1.

Послідовність дій по створенню форми - "Складний цикл"

№ п/п	Операція	Властивість об'єкта	Значення властивості
1	Дати ім'я електронній формі.	Name	<i>Form1</i>
		Caption	<i>Складний цикл</i>

2	Створити текстове вікно для виведення розрахункових даних (початок інтервалу) (інструмент TextBox)	Name	<i>Text1</i>
		Text	<i>Складний цикл</i>
		Multline	<i>True</i>
		ScrollBars	<i>2-Вертикально</i>
3	Створити текстове вікно для введення назви ВНЗ, факультету, групи, прізвища студента (інструмент TextBox)	Name	<i>Text2</i>
		Multline	<i>True</i>
		Text	Порожньо
4	Створити етикетку (інструмент Label)	Name	<i>Label1</i>
		Caption	<i>Обчислити значення функції</i>
5	Створити у формі для ілюстрації формули $Y = \sum_{n=1}^{10} x^n / n$, якщо $X < 0,55$ (інструмент OLE)	Name	OLE1
6	Створити у формі для ілюстрації формули $Y = \frac{X^2}{3} \prod_{k=1}^{15} (1 + x/k)$, якщо $X \geq 0,55$ (інструмент OLE)	Name	OLE2
7	Створити командну кнопку для виконання програми (інструмент CommandButton)	Name	<i>Command1</i>
		Caption	<i>Обчислення</i>
8	Створити командну кнопку для очищення текстового вікна (інструмент CommandButton)	Name	<i>Command2</i>
		Caption	<i>Очищення вікна</i>
9	Створити командну кнопку для виходу з програми (інструмент CommandButton)	Name	<i>Command3</i>
		Caption	<i>Вихід</i>

3) Робочий вид екранної форми. Остаточна екранна форма має вигляд, приведений на рис. 8.3

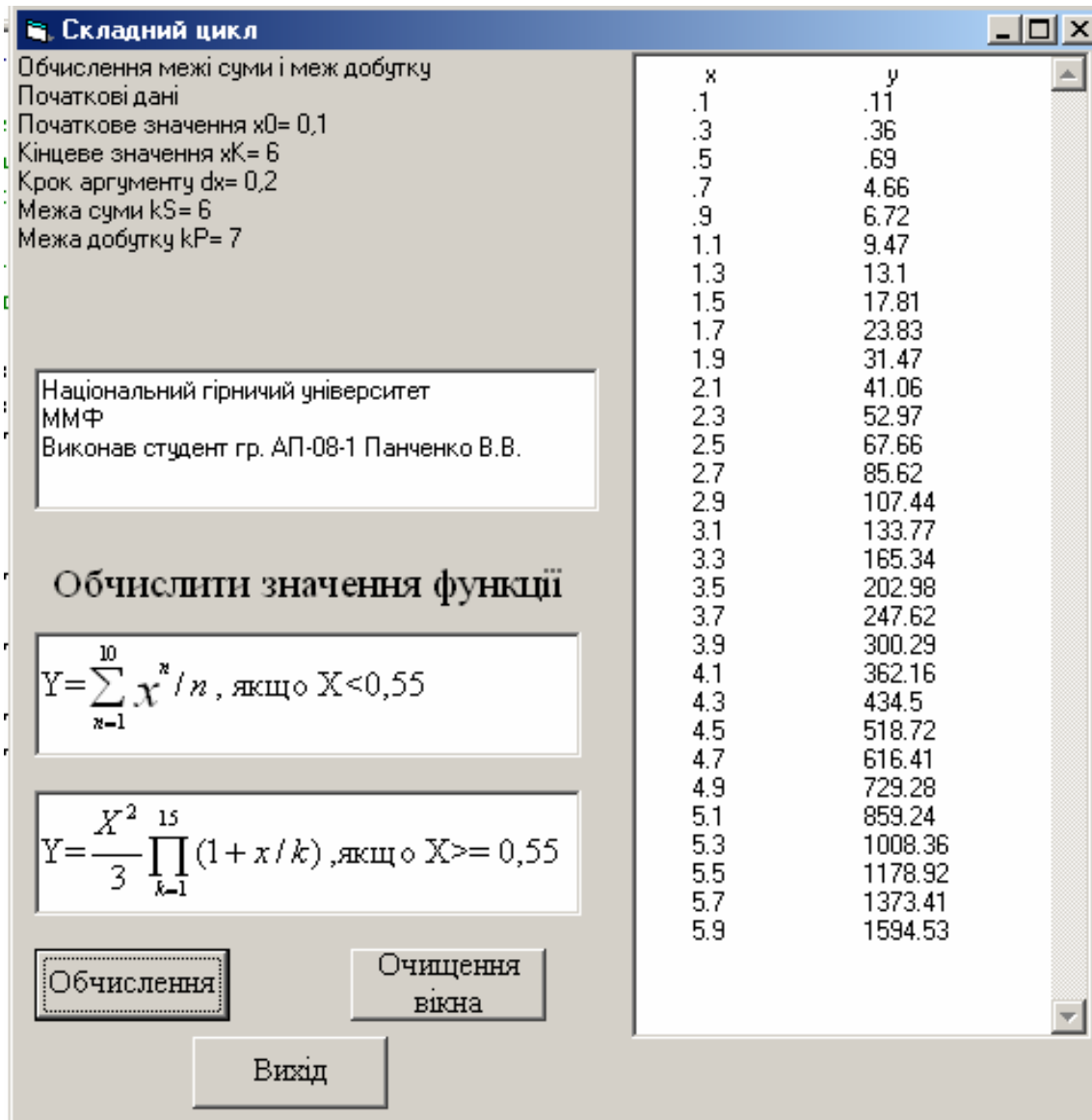


Рис. 8.3. Остаточний вигляд форми "Складний цикл"

4) Створення програмного коду. Повний програмний код приводиться далі:

```

Private Sub Command1_Click()
Cls
'Обчислення суми і добутку ряду чисел
'x - аргумент функції, y - функція
'x0, xk, dx - початкове, кінцеве і крок аргументу
'kS - межа суми, kP - межа добутку
'k,n - параметри циклів
'Виконав студент гр. АП-08-1
Dim x, y, x0, Xk, dx As Single
Dim ks, kp As Integer
Dim xs, ys, z As String
x0 = Val(InputBox("Введіть початкове значення x0="))
Print "Обчислення межі суми і меж добутку"
Print "Початкові дані"
Print "Початкове значення x0="; x0
Xk = Val(InputBox("Введіть кінцеве значення xK="))
Print "Кінцеве значення xK="; Xk
dx = Val(InputBox("Введіть крок аргументу dx="))
Print "Крок аргументу dx="; dx
ks = Val(InputBox("Введіть межу суми kS="))
kp = Val(InputBox("Введіть межу добутку kP="))
Print "Межа суми kS="; ks
Print "Межа добутку kP="; kp
' Побудова шапки для таблиці
Text1.Text = ("          x                               y") & Chr(13) & Chr(10)
For x = x0 To Xk Step dx
    If x < 0.55 Then
        y = 0
        ' Організація вкладеного циклу для обчислення функції y=y1(x)
        For n = 1 To ks
            y = y + x ^ n / n
            xs = Str(Format$(x, "00.00"))
            ys = Str(Format$(y, "00.00"))
        Next n
    Else
        y = 1
        For k = 1 To kp
            y = y * (1 + x / k)
            xs = Str(Format$(x, "00.00"))
            ys = Str(Format$(y, "00.00"))
        Next k
        y = x * x * y / 3
    End If
    Text1.Text = Text1.Text + "          " + xs + Chr(9) _
        + "          " + ys + Chr(13) + Chr(10)
Next x
End Sub
' Процедура очищення вікна
Private Sub Command2_Click()
Text1.Text = ""
End Sub
' Процедура виходу з програми
Private Sub Command3_Click()
End
End Sub

```

9. Лабораторна робота № 7

Завдання: Розробити програму для побудови графіка функції

$$y = \sqrt{x} e^{\sin(\pi x)}$$

У програмі виконати наступне:

- представити зображення графіка функції в заданому інтервалі;
- вивести на поверхню форми мінімальні і максимальні значення змінних x і y ;
- побудувати координатні осі і масштабну сітку;
- виконати автоматичне оцифрування шкали;
- на поверхні форми з правого боку побудувати зображення державного прапора України.

Виконання:

1) Створити екранну форму згідно рис. 9.1.

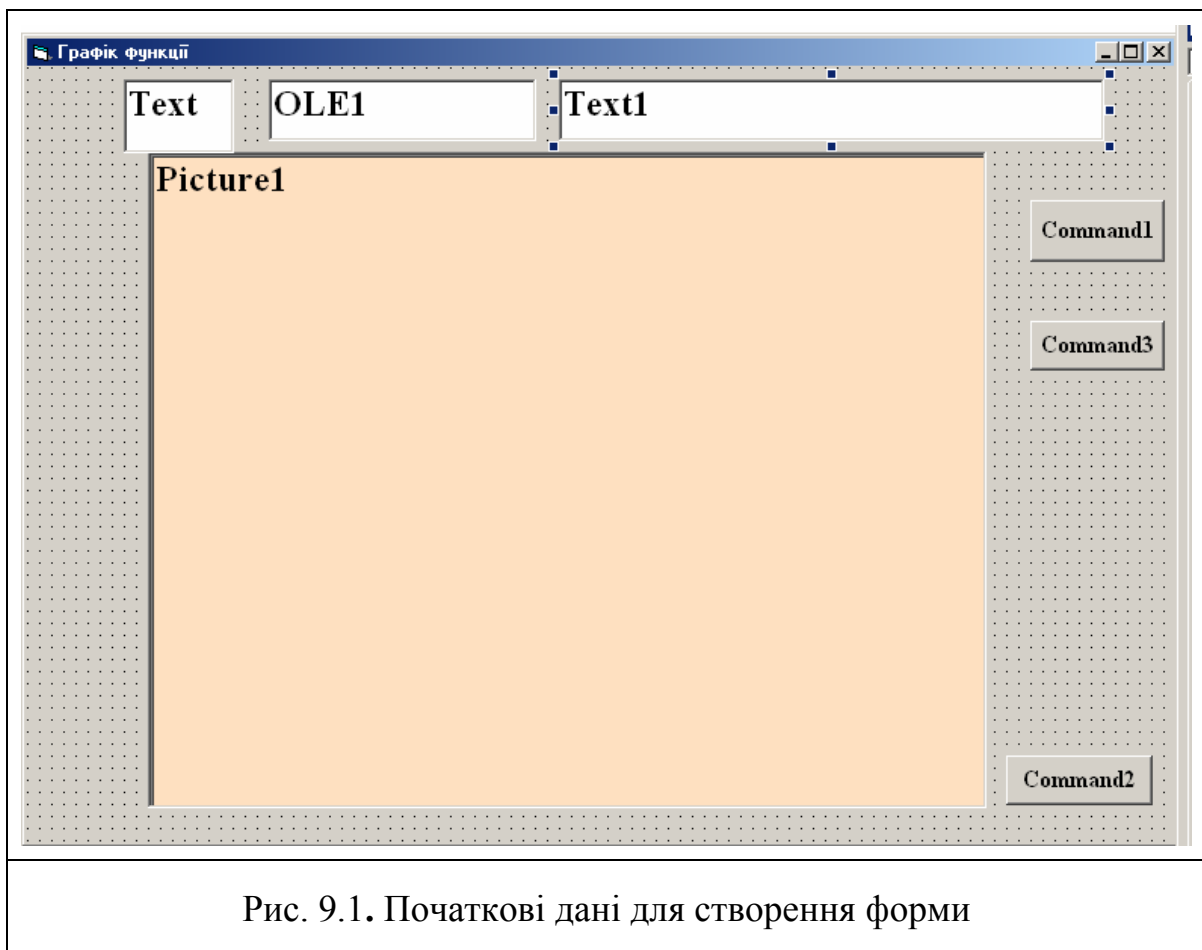


Рис. 9.1. Початкові дані для створення форми

Послідовність дій по створенню форми "Вибір функції" приведена в таблиці 9.1

Послідовність дій по створенню форми - "Графік функції"

№ п/п	Операція	Властивість об'єкта	Значення властивості
1	Дати ім'я електронній формі.	Name	<i>Form1</i>
		Caption	<i>Графік функції</i>
2	Створити текстове вікно для відображення значень гальмівного шляху (інструмент TextBox)	Name	<i>Text1</i>
		Text	<i>Виконав студент гр. АП</i>
		Multiline	<i>True</i>
3	Створити текстове вікно для відображення значень гальмівного шляху (інструмент TextBox)	Name	<i>Text2</i>
		Text	<i>Графік функції</i>
		Multiline	<i>True</i>
	Створити командну кнопку для виконання розрахунку гальмівного шляху(інструмент CommandButton)	Name	<i>Command1</i>
		Caption	<i>Запуск програми</i>
	Створити командну кнопку для виходу з програми (інструмент CommandButton)	Name	<i>Command3</i>
		Caption	<i>Вихід</i>
	Створити командну кнопку для очищення вікон (інструмент CommandButton)	Name	<i>Command2</i>
		Caption	<i>Очищення</i>
	Створити вікно для побудови графіка інтенсивності пасажиропотоку (інструмент PictureBox)	Name	<i>Picture1</i>
		AutoRedraw	<i>True</i>
	Создать окно для вставки вида функциональной зависимости Y (інструмент OLE)	Name	<i>OLE1</i>

Остаточною екранною формою має вигляд, приведений на рис. 9.2

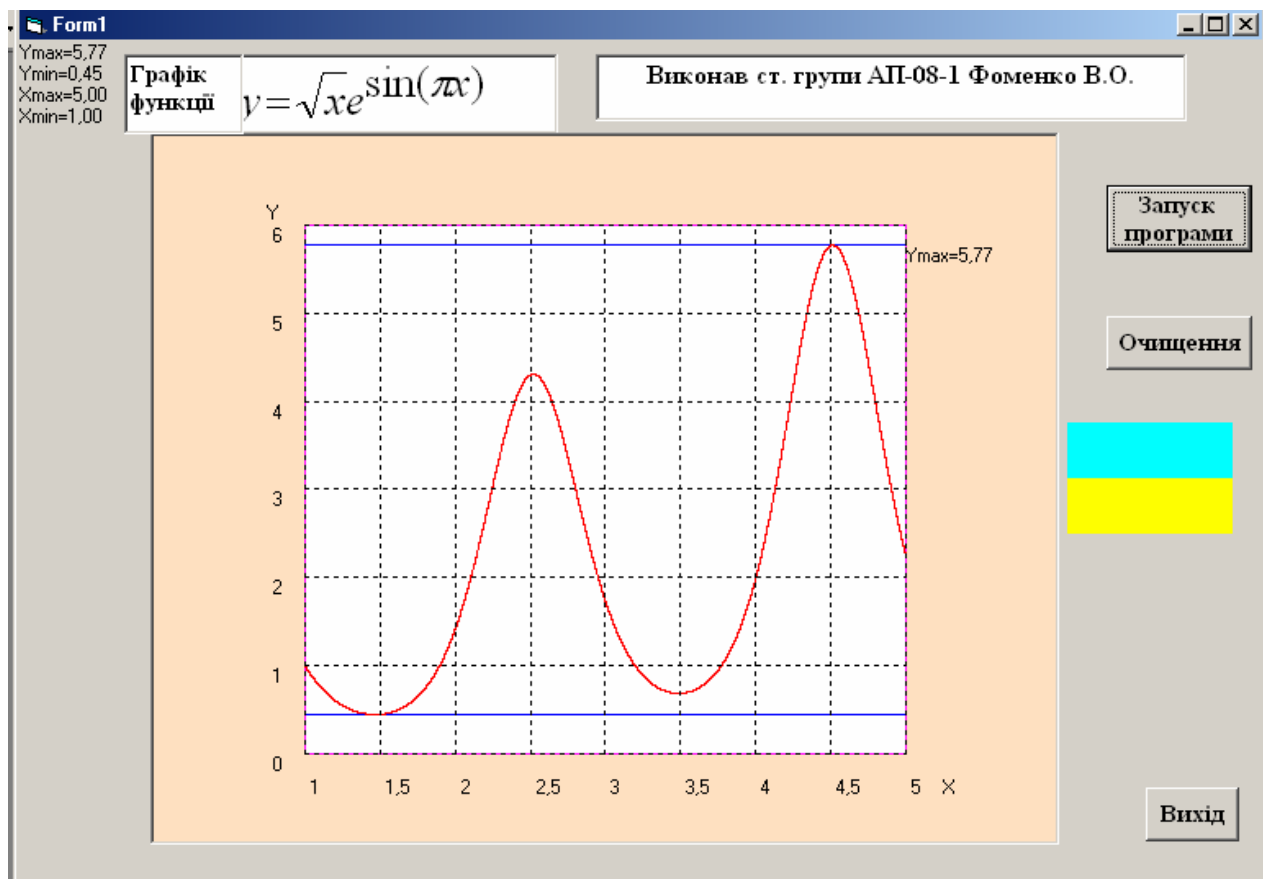


Рис. 9.2. Остаточний вигляд форми для побудови графіку

3) Створити програмний код. Повний програмний код приводиться далі:

' Процедура графічних побудов

Sub graf_fun()

Dim ymin As Single

Dim ymax As Single

ymin = -100000000#

ymax = 1000000000#

' Введення мінімальних і максимальних значень аргументу

xmin = Val(InputBox("ввод XMIN="))

xmax = Val(InputBox("ввод XMAX="))

' Організація циклу для знаходження в інтервалі $x_{min} \leq x \leq x_{max}$

' мінімальних і максимальних значень функції Y

For x = xmin To xmax Step 0.001

y = (Sqr(x)) * Exp(Sin(3.14159 * x))

If y > ymin Then

ymax = y

ymin = y

End If

If y < ymax Then

ymax = y

```

    ymax = y
End If
Next x ' кінець циклу
' Друк на поверхні форми мінімальних і максимальних значень змінних x і y
Print "Ymax="; (Format(ymax, "0.00"))
Print "Ymin="; (Format(ymin, "0.00"))
Print "Xmax="; (Format(xmax, "0.00"))
Print "Xmin="; (Format(xmin, "0.00"))
ymaxlin = Int(ymax) + 1
yminlin = Int(ymin)
dx = xmax - xmin
dy = ymaxlin - yminlin
' Масштабування площі вікна малювання
Picture1.Scale ((xmin - 1), (ymaxlin + 1))-((xmax + 1), (yminlin - 1))
' Закрашення в білий колір прямокутної зони кривої графіка
Picture1.Line (xmin, yminlin)-(xmax, ymaxlin), vbWhite, BF
' Малювання прямокутника графічної зони побудови графіка
Picture1.Line (xmin, yminlin)-(xmax, ymaxlin), vbMagenta, B
Picture1.DrawStyle = vbSolid
' Малювання горизонтальної лінії, відповідної максимальному
' значенню функції Y
Picture1.Line (xmin, ymax)-(xmax, ymax), vbBlue
Picture1.Print "Ymax="; (Format(ymax, "0.00"))
' Малювання горизонтальної лінії, відповідної мініимальному
' значенню функції Y
Picture1.Line (xmin, ymin)-(xmax, ymin), vbBlue
' Організація циклу для побудови графіка
For x = xmin To xmax Step 0.0001
    y = (Sqr(x)) * Exp(Sin(3.14159 * x))
    Picture1.PSet (x, y), vbRed
Next x ' кінець циклу
Picture1.PSet (xmax, ymax)
Picture1.PSet (xmin, ymin)
Picture1.PSet (xmax, ymin)
Picture1.PSet (xmin, ymin)
Picture1.CurrentX = xmin - 0.25
Picture1.CurrentY = ymaxlin + 0.25
Picture1.Print "Y" ' позначення осі ординат Y
Print
'-----
' Визначення кроку масштабної сітки уздовж осі ординат Y
If dy <= 1 Then
    sty = 0.1
ElseIf dy <= 2 Then
    sty = 0.2

```

```

ElseIf dy <= 5 Then
    sty = 0.5
ElseIf dy <= 10 Then
    sty = 1
ElseIf dy <= 20 Then
    sty = 2
ElseIf dy <= 50 Then
    sty = 5
ElseIf dy <= 100 Then
    sty = 10
ElseIf dy <= 500 Then
    sty = 50
ElseIf dy <= 1000 Then
    sty = 100
ElseIf dy <= 2000 Then
    sty = 200
ElseIf dy <= 5000 Then
    sty = 500
ElseIf dy <= 10000 Then
    sty = 1000
End If
For zx = yminlin To ymaxlin Step sty
    Picture1.DrawStyle = vbDot
    Picture1.Line (xmin, zx)-(xmax, zx)
    Picture1.CurrentX = xmin - 0.25
    Picture1.CurrentY = zx
    Picture1.Print zx
Next zx
Picture1.CurrentX = xmax + 0.25
Picture1.CurrentY = yminlin - 0.25
Picture1.Print "X" ' позначення осі абсцис X
Print
' Визначення кроку масштабної сітки уздовж осі абсцис X
If dx <= 1 Then
    stx = 0.1
ElseIf dx <= 2 Then
    stx = 0.2
ElseIf dx <= 5 Then
    stx = 0.5
ElseIf dx <= 10 Then
    stx = 1
ElseIf dx <= 20 Then
    stx = 2
ElseIf dx <= 50 Then
    stx = 5

```



```

ElseIf dx <= 100 Then
    stx = 10
ElseIf dx <= 500 Then
    stx = 50
ElseIf dx <= 1000 Then
    stx = 100
ElseIf dx <= 2000 Then
    stx = 200
ElseIf dx <= 5000 Then
    stx = 500
ElseIf dx <= 10000 Then
    stx = 1000
End If
' Організація циклу для малювання масштабної сітки
For zy = xmin To xmax Step stx
    Picture1.Line (zy, yminlin)-(zy, ymaxlin)
    Picture1.CurrentX = zy
    Picture1.CurrentY = yminlin - 0.25
    Picture1.Print zy
Next zy ' кінець циклу
' Побудова зображення державного прапора України
Line (9600, 3500)-(11100, 4000), vbCyan, BF
Line (9600, 4000)-(11100, 4500), vbYellow, BF
End Sub

' Процедура запуску програми
Private Sub Command1_Click()
    Call graf_fun
End Sub

' Процедура виходу з програми
Private Sub Command2_Click()
    End
End Sub

' Процедура очищення вікон
Private Sub Command3_Click()
    Picture1.Cls
    Form1.Cls
End Sub

```

10. Лабораторна робота № 8. "Розрахунок пасажиропотоку на маршруті"

Завдання. На основі хронометричних даних встановлено, що інтенсивність пасажиропотоку (кількість пасажирів, що перевозяться в одиницю часу) в певний час доби обчислюється за формулою:

$$S=120 + 1750 \cdot e^{-x} \cdot \text{SIN}(x), \text{ де } X=3,14 \cdot (t-6) / 9;$$

t - час доби, в якому характеризується пасажиропотік.

Кількість пасажирів що перевозяться за певний час обчислюється за формулою

$$S_{\text{pas}} = \int_{t1}^{t2} S dt, \text{ де } t1 \text{ и } t2 - \text{ час почала і кінця інтервалу.}$$

Розробити програму в якій можна розрахувати кількість пасажирів, що перевозяться в певному інтервалі часу доби (наприклад, з 9 до 12 годин дня).

Розрахувати також загальну кількість пасажирів, що перевозяться, за тиждень в інтервалі часу з 6 годин до 15 годин. Побудувати графік інтенсивності пасажиропотоку в інтервалі цього ж часу.

Виконання:

1) Створити початкову екранну форму згідно рис. 10.1. На поверхні форми приведені імена і назви компонентів, прийняті за умовчанням. Надалі імена кожного з компонентів не змінний, а їх назви приймаємо відповідно до виконуваних і необхідних дій.

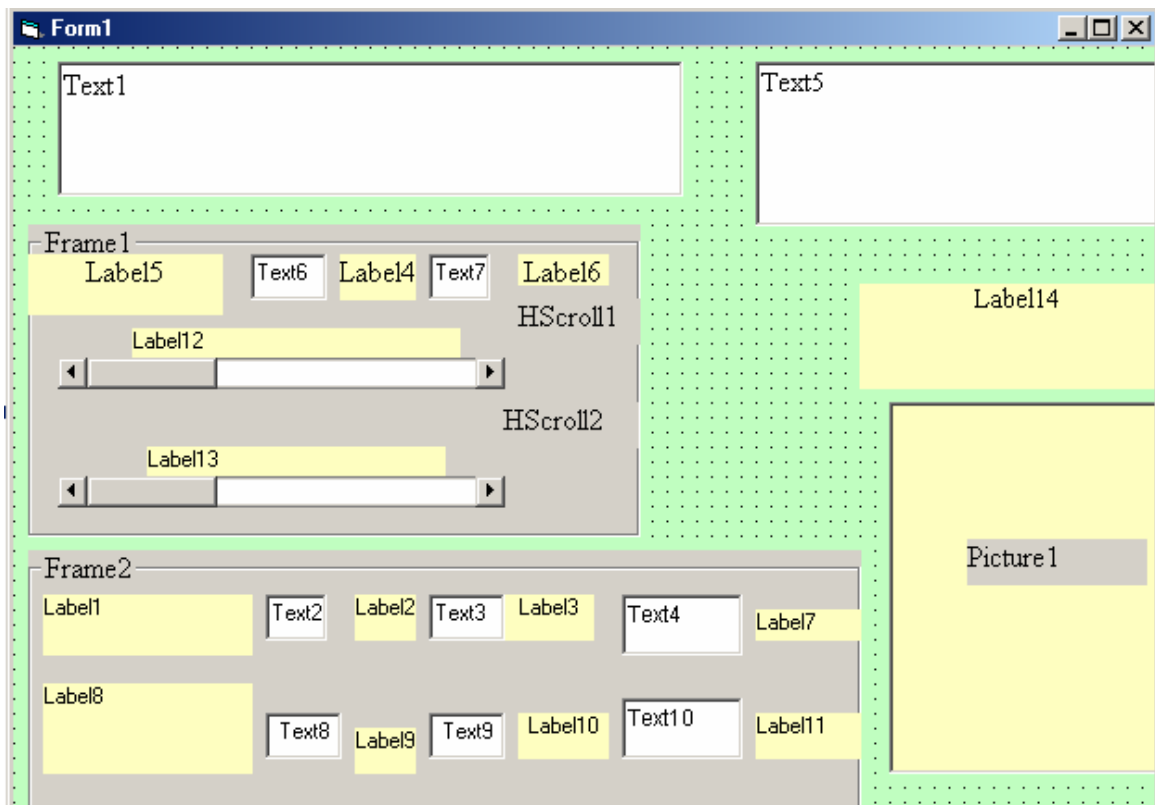


Рис. 10.1. Початкові дані для створення форми

2) Послідовність дій по створенню форми "Розрахунок пасажиропотоку" приведена в таблиці 1

Таблиця 10.1.

Послідовність дій по створенню форми – “Розрахунок пасажиропотоку”

№ п/п	Операція	Властивість об'єкта	Значення властивості
1	Дати ім'я електронній формі.	Name	<i>Form1</i>
		Caption	<i>Розрахунок_пасажиропотоку</i>
2	Створити рамку групи інструментів для введення початкових даних (інструмент Frame)	Name	<i>Frame1</i>
		Caption	<i>Початкові дані</i>
3	Створити рамку групи інструментів для виведення розрахункових даних (інструмент Frame)	Name	<i>Frame2</i>
		Caption	<i>Розрахункові дані</i>
4	Створити текстове вікно для назви лабораторної роботи (інструмент TextBox)	Name	<i>Text1</i>
		Text	<i>Розрахунок пасажиропотоку на тролейбусному маршруті</i>
		Multiline	<i>True</i>
5	Створити текстове вікно для виведення розрахункових даних (початок інтервалу) (інструмент TextBox)	Name	<i>Text2</i>
		Text	Порожньо
6	Створити текстове вікно для виведення розрахункових даних (кінець інтервалу) (інструмент TextBox)	Name	<i>Text3</i>
		Text	Порожньо
7	Створити текстове вікно для виведення розрахункових даних (кількість пасажирів за інтервал руху) (інструмент TextBox)	Name	<i>Text4</i>
		Text	Порожньо
8	Створити текстове вікно для введення назви ВНЗ, факультету, групи, прізвища студента (інструмент TextBox)	Name	<i>Text5</i>
		Text	Порожньо
		Multiline	<i>True</i>

9	Створити текстове вікно для введення початкових даних (інструмент TextBox)	Name	<i>Text6</i>
		Text	Порожньо
10	Створити текстове вікно для введення початкових даних (інструмент TextBox)	Name	<i>Text7</i>
		Text	Порожньо
11	Створити текстове вікно для виведення розрахункових даних тиждень (інструмент TextBox)	Name	<i>Text8</i>
		Text	<i>6</i>
12	Створити текстове вікно для виведення розрахункових даних за тиждень (інструмент TextBox)	Name	<i>Text9</i>
		Text	<i>15</i>
13	Створити текстове вікно для виведення розрахункових даних тиждень (інструмент TextBox)	Name	<i>Text10</i>
			Порожньо
14	Створити лінійку прокручування для введення початкового часу інтервалу руху (інструмент HScrollBar)	Name	<i>HScroll1</i>
		Large Change	<i>1</i>
		Max	<i>15</i>
		Min	<i>6</i>
15	Створити лінійку прокручування для введення кінцевого часу інтервалу руху (інструмент HScrollBar)	Name	<i>HScroll2</i>
		Large Change	<i>1</i>
		Max	<i>15</i>
		Min	<i>6</i>
16	Створити етикетку (інструмент Label)	Name	<i>Label1</i>
		Caption	<i>Число перевезених пасажирів з</i>
17	Створити етикетку (інструмент Label)	Name	<i>Label2</i>
		Caption	<i>до</i>
18	Створити етикетку (інструмент Label)	Name	<i>Label3</i>
		Caption	<i>годин складає</i>
19	Створити етикетку (інструмент Label)	Name	<i>Label4</i>
		Caption	<i>до</i>

20	Створити етикетку (інструмент Label)	Name	<i>Label5</i>
		Caption	<i>Інтервал часу з</i>
21	Створити етикетку (інструмент Label)	Name	<i>Label6</i>
		Caption	<i>годин</i>
22	Створити етикетку (інструмент Label)	Name	<i>Label7</i>
		Caption	<i>пасажирів</i>
23	Створити етикетку (інструмент Label)	Name	<i>Label8</i>
		Caption	<i>Число перевезених пасажирів за тиждень в інтервалі часу з</i>
24	Створити етикетку (інструмент Label)	Name	<i>Label9</i>
		Caption	<i>до</i>
25	Створити етикетку (інструмент Label)	Name	<i>Label9</i>
		Caption	<i>Інтервал часу з</i>
26	Створити етикетку (інструмент Label)	Name	<i>Label10</i>
		Caption	<i>годин складає</i>
27	Створити етикетку (інструмент Label)	Name	<i>Label11</i>
		Caption	<i>пасажирів</i>
28	Створити етикетку (інструмент Label)	Name	<i>Label12</i>
		Caption	<i>Введить початковий час інтервалу</i>
29	Створити етикетку (інструмент Label)	Name	<i>Label13</i>
		Caption	<i>Введить кінцевий час інтервалу</i>
30	Створити етикетку (інструмент Label)	Name	<i>Label14</i>
		Caption	<i>Графік інтенсивності пасажиропотоку з 6 до 15 годин, люд/год</i>

31	Створити вікно для побудови графіка інтенсивності пасажиропотоку (інструмент PictureBox)	Name	<i>Picture1</i>
		AutoRedraw	<i>True</i>
32	Створити командну кнопку для виконання розрахунку інтенсивності грузопотоку (інструмент CommandButton)	Name	<i>Command1</i>
		Caption	<i>Розрахунок пасажиропотоку</i>
33	Створити командну кнопку для очищення полів (інструмент CommandButton)	Name	<i>Command2</i>
		Caption	<i>Очистити вікна</i>
34	Створити командну кнопку для виходу з програми (інструмент CommandButton)	Name	<i>Command3</i>
		Caption	<i>Вихід</i>

Остаточно екранна форма має вигляд, приведений на рис. 10.2

3) Створити програмний код. Повний програмний код приводиться далі:

' Процедура побудови графіка інтенсивності

' пасажиропотоку залежно від часу доби

Sub SPicture1()

Cls *' очищення поверхні вікна*

Picture1.Scale (4, 700)-(15, -50) *' задание размеров графического поля и
' направления 'координатних осей*

s2 = 0

Max = -1000

' Формирование цикла для построения графика функции интенсивности

' пасажиропотока в зависимости от времени суток

For t2 = 6 To 15

X2 = 3.14152 * (t2 - 6) / 9

s2 = 120 + 1750 * Exp(-X2) * Sin(X2)

X21 = 3.14152 * ((t2 + 1) - 6) / 9

s21 = 120 + 1750 * Exp(-X21) * Sin(X21)

If s21 > Max Then Max = s21

Picture1.Line (t2, s2)-((t2 + 1), s21), vbRed

Picture1.Line (t2, 120)-(t2, s2), vbGreen

Next t2

Picture1.Line (6, 120)-(15, 120), vbBlue *'ось X*

Picture1.Line (6, 0)-(6, 700), vbBlue *'ось Y*

Picture1.Line (6, Max)-(8, Max), vbBlue ' лінія, що позначає Y_{max}

Розрахунок пасажиропотоку на тролейбусному маршруті

НГУ
ММФ
АП-07-1
Петренко В.В.

Початкові дані
Інтервал часу з 7 до 15 годин
Введіть початковий час інтервалу
Введіть кінцевий час інтервалу

Розрахункові дані
Число перевезених пасажирів з 7 до 15 годин складає 3100 пасажирів
Число перевезених пасажирів за тиждень в інтервалі часу з 6 до 15 годин складає 18210 пасажирів

Графік інтенсивності пасажиропотоку з 6 до 15 годин, люд/год

679
220
6 8 10 12 14
час доби

Розрахунок пасажиропотоку Очистити вікна Вихід

Рис. 10.2. Остаточний вигляд форми для розрахунку пасажиропотоку

' Оцифрування шкали X

Picture1.PSet (6, 100)

Picture1.Print "6"

Picture1.PSet (8, 100)

Picture1.Print "8"

Picture1.PSet (10, 100)

Picture1.Print "10"

Picture1.PSet (12, 100)

Picture1.Print "12"

Picture1.PSet (14, 100)

Picture1.Print "14"

Picture1.PSet (6, 45)

Picture1.Print " час доби"

Picture1.PSet (4.3, 170)

' Позначення мінімальних і максимальних значень інтенсивності

```
' пасажиропотоку
Picture1.Print "220"
Picture1.PSet (4.3, (25 + Max))
Picture1.Print Int(Max)
End Sub
```

```
' Процедура очищення вікон
Private Sub Command2_Click()
Text6.Text = ""
Text7.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text10.Text = ""
Text5.Text = ""
Picture1.Cls
End Sub
```

```
' Процедура виходу з програми
Private Sub Command3_Click()
End
End Sub
```

```
' Процедури створення і роботи з лінійками прокручування для введення  
' початкового і кінцевого часу інтервалу
```

```
Private Sub HScroll2_Scroll()
HScroll2_Change
End Sub
```

```
Private Sub HScroll2_Change()
lb2 = HScroll2
Text7.Text = Str(lb2)
End Sub
```

```
Private Sub HScroll1_Scroll()
HScroll1_Change
End Sub
```

```
Private Sub HScroll1_Change()
lb1 = HScroll1 - 1
Text6.Text = Str(lb1)
End Sub
```

```
' Процедура використання кнопки, Command1 для виконання  
' розрахунку обчислення пасажиропотоків
```



```

Private Sub Command1_Click()
    enter1 = Chr(13) + Chr(10)
    vus = InputBox("Введіть назву навчального закладу, в якому ви вчитися")
    fak = InputBox("Введіть назву факультету")
    grup = InputBox("Введіть назву групи")
    fam = InputBox("Введіть своє прізвище та ініціали ")
    Text5.Text = vus + enter1 + fak + enter1 + grup + enter1 + fam
    S = 0: k = 0: s1 = 0
    For i = 1 To 5 ' створення вкладеного циклу для обчислення пасажиропотоку
        ' за тиждень у заданому інтервалі
        For t = 7 To 15
            x = 3.14152 * (t - 6) / 9
            S = S + 120 + 1750 * Exp(-x) * Sin(x)
        Next t
    Next i
    Text2.Text = Text6.Text
    Text3.Text = Text7.Text
    l1 = Val(Text6.Text) + 1
    l2 = Val(Text7.Text)
    For t1 = l1 To l2 Step 1 ' початок циклу; створення циклу для обчислення
        ' пасажиропотоку у заданому інтервалі часу доби
        k = k + 1
        X1 = 3.14152 * (t1 - 6) / 9
        s1 = s1 + 120 + 1750 * Exp(-X1) * Sin(X1)
    Next t1 ' кінець циклу
    Cls
    Text4.Text = Str(Int(s1))
    Text10 = Str(Int(S))
    SPicture1 ' використання процедури SPicture1 для побудови графіка
        ' інтенсивності пасажиропотоку
End Sub

```

Література

1. Войтюшенко, Інформатика і комп'ютерна техніка: Навч. пос. з баз. підготовки для студ. екон. і техн. спеціальностей ден. і заоч. форм навчання / Н.М. Войтюшенко, А.І. Остапець. – К.: Центр навчальної літератури, 2006 – 568 с.
2. Вычислительная техника и программирование: учебник для техн. вузов / А.В. Петров, В.Е. Алексеев, А.С. Ваулин и др.; под ред. А. В. Петрова. – М.: Высшая школа, 1990. – 479 с.: ил.
3. Глинський, Практикум з інформатики: навч. посібник 8-е вид. переробл. і доп. / Я.М. Глинський. – Л.: Деол, СПД Глинський, 2005. – 296 с.: іл..
4. Глинський, Бейсик, Visual Basic і VBA. 4-те вид.перероб. і доп. / Я.М. Глинський, В.Є. Анохін, В.А. Ряжська. – Л.: Деол, СПД Глинський, 2004. – 160 с.
5. Глушаков, Программирование на Visual Basic 6.0: учеб. курс / С.В. Глушаков, С.А. Сурядный; Худож. оформ. А.С. Юхтман. – Х.: Фолио, 2004. – 497 с.
6. Дибкова, Інформатика і комп'ютерна техніка: навч. посіб. / Л.М. Дибкова. – вид. 2-ге, перероб. і доп. – К.: Академвидав, 2005. – 416 с.: іл.
7. Информатика. Базовый курс. / под ред. С.В. Симоновича. – 2-е изд. – С.Пб.: Питер. 2007. – 640 с.: ил.
8. Ковтанюк, Самовчитель роботи на ПК / Ю.С. Ковтанюк, Ю.О. Шпак. –К.: МК-Прес, 2005. – 544 с.: іл.
9. Культин, Visual Basic. Освой самостоятельно / Н.Б. Культин – С.Пб.: БХВ-Петербург, 2005. – 480 с. ил.
10. Сафронов И. К. Visual Basic в задачах и примерах. – С.Пб.: БХВ-Петербург, 2007. - 400 с.: ил.