

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Стець О.В.

ІНФОРМАТИКА 2. ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ

Комп'ютерний практикум

Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для здобувачів ступеня бакалавр
за спеціальності 051 Економіка

Укладач: О.В. Стець

Електронне мережеве навчальне видання

Київ
КПІ ім. ІГОРЯ СІКОРСЬКОГО
2025

УДК 004.9, 330

Укладач *Стець Олена Вікторівна*, канд. фіз.-мат. наук., доцент

Рецензент *Петренко Катерина Валеріївна*, канд. екон. наук, доцент кафедри міжнародної економіки, КПІ ім. Ігоря Сікорського

Відповідальний редактор *Бояринова Катерина Олександрівна*, докт. екон. наук, професор

*Гриф надано Методичною радою КПІ ім. Ігоря Сікорського
(протокол № 5 від 06.03.2025 р.)
за поданням вченої ради факультету менеджменту та маркетингу
(протокол № 7 від 30.01.2025р.)*

Інформатика 2. Технології програмування: Комп'ютерний практикум [Електронний ресурс] : навч. посіб. для здобувачів ступеня бакалавр за спец. 051 Економіка / КПІ ім. Ігоря Сікорського; уклад.: О.В.Стець. – Електрон. текст. дані (1 файл). – Київ : КПІ ім. Ігоря Сікорського, 2025. – 105 с.

Навчально наочний посібник з дисципліни «Інформатика 2. Технології програмування» відповідає навчальній програмі дисципліни «Інформатика 2. Технології програмування» для здобувачів ступеня магістра денної та заочної форм навчання за спеціальністю 051 Економіка. У навчальному посібнику показано застосування теоретичного матеріалу до розв'язування поставлених практичних завдань відповідно до завдань комп'ютерних практикумів.

УДК 004.9, 330

Реєстр. № НП 24/25-242. Обсяг 2,2 авт. арк.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
проспект Берестейський, 37, м. Київ, 03056
<https://kpi.ua>

Свідоцтво про внесення до Державного реєстру видавців, виготовлювачів і розповсюджувачів видавничої продукції ДК № 5354 від 25.05.2017 р.

© КПІ ім. Ігоря Сікорського, 2025

ЗМІСТ

ВСТУП	4
Комп'ютерний практикум №1	6
Комп'ютерний практикум №2	16
Комп'ютерний практикум №3	32
Комп'ютерний практикум №4	37
Комп'ютерний практикум №5	43
Комп'ютерний практикум №6	49
Комп'ютерний практикум №7	59
Комп'ютерний практикум №8	64
Комп'ютерний практикум №9	81
Список використаних джерел	99
ДОДАТОК А Налаштування NotePad++	100
ДОДАТОК Б Робота з VB в середовищі MS Office.....	105

ВСТУП

Сучасний фахівець, що працює в умовах цифрової економіки, повинен володіти не лише ґрунтовними теоретичними знаннями з програмування та інформаційних технологій, а й уміти застосовувати їх на практиці для вирішення конкретних бізнес-задач. Саме реальні навички автоматизації офісних процесів, ефективного керування даними та створення функціональних вебресурсів визначають конкурентоспроможність спеціаліста на сучасному ринку праці. Здатність швидко перетворювати теоретичні знання на дієві програмні рішення є ключовим індикатором професійної підготовки. Методичні вказівки до комп'ютерних практикумів з дисципліни «Інформатика 2. Технології програмування» розроблені з метою надання студентам інструментарію для набуття стійких практичних умінь. Основний акцент зроблено на двох критично важливих напрямках:

- Основи веброботи: Застосування мови розмітки HTML та базових вебтехнологій (CSS) для ефективного структурування, форматування та публікації інформаційних ресурсів в Інтернеті.
- Автоматизація офісних процесів: Використання мови програмування Visual Basic for Applications (VBA) для створення макросів, спеціалізованих функцій та форм, що дозволяють автоматизувати рутинні та складні аналітичні завдання безпосередньо в середовищі програмного пакету MS Office (зокрема MS Excel).

Запропоновані практикуми мають чітку поетапну структуру, що забезпечує послідовне формування професійних навичок: від розуміння основних принципів алгоритмізації та створення простих макросів до реалізації складніших алгоритмів керування даними у VBA, від роботи з базовою HTML-розміткою до професійного структурування та візуалізації інформації на вебсторінках.

Виконання цих практичних завдань дозволить студентам:

- Закріпити теоретичний матеріал, отриманий у лекційному курсі.



- Створювати власні програмні рішення для підвищення продуктивності праці та мінімізації помилок.
- Будувати та адмініструвати базові вебресурси для ефективного представлення інформації цільовій аудиторії.
- Сформулювати цілісне уявлення про можливості сучасних інформаційних технологій у їхній майбутній професійній діяльності.

Особливу увагу в практикумах приділено питанням ефективності коду, безпеки даних та етичного використання інформаційних технологій, що є необхідною умовою роботи в сучасному цифровому середовищі. Ці методичні вказівки є надійним супутником студента на шляху перетворення теоретичних знань на практично застосовні компетенції.

Комп'ютерний практикум №1

Створення HTML документів

Робота складається з прикладів та самостійного завдання.

Кожне завдання створюються окремим файлом в  Блокноті (NotePad) або  NotePad++

Файли зберігаються із розширенням **.html**

Кодування HTML-сторінки необхідно вказувати для того, щоб web-браузер коректно відтворював текст на сторінці. Якщо браузер «не вгадає» кодування, то замість тексту будуть відображатися ієрогліфи.

Щоб надати браузеру інформацію про кодування HTML-сторінки, необхідно всередині тега <head>використати тег:

<meta charset="ім'я кодування">

Найбільш розповсюджене сучасне кодування — **utf-8**

** Крім того, слід зазначити, що більшість прикладів розроблено без 100%-го використання css-правил; використовуються теги, як більш раніх версій HTML , так і більш сучасних (для широти розуміння 😊)*

<p><i>CSS Cascading Style Sheets (каскадні листи стилів)</i> – це набір правил оформлення і форматування, який може бути застосований до різних елементів сторінки. Застосовуючи CSS, ви можете один раз описати властивості елементів і визначити цей опис як стиль, а надалі просто вказувати, що елемент, який ви бажаєте оформити відповідним чином, повинен прийняти властивості стилю, описаного Вами.</p>
--

Основна концепція CSS - форматування документа повинне бути відділене від його змісту

Завдання

1. Приклад роботі зі шрифтами

1.1. Дуже популярні, але на жаль застарілі теги роботи зі шрифтами

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<title> Робота з font </title>
</head>

<body>
  <font size=4> <b> Текст </b></font><br>
  <font size=7> <i> Текст </i></font><br>
  <font size=9 color=green> <u> Текст </u></font><br>
</body>
</html>
```

1.2. *В сучасній версії HTML все, що стосується форматування замінено на **style***

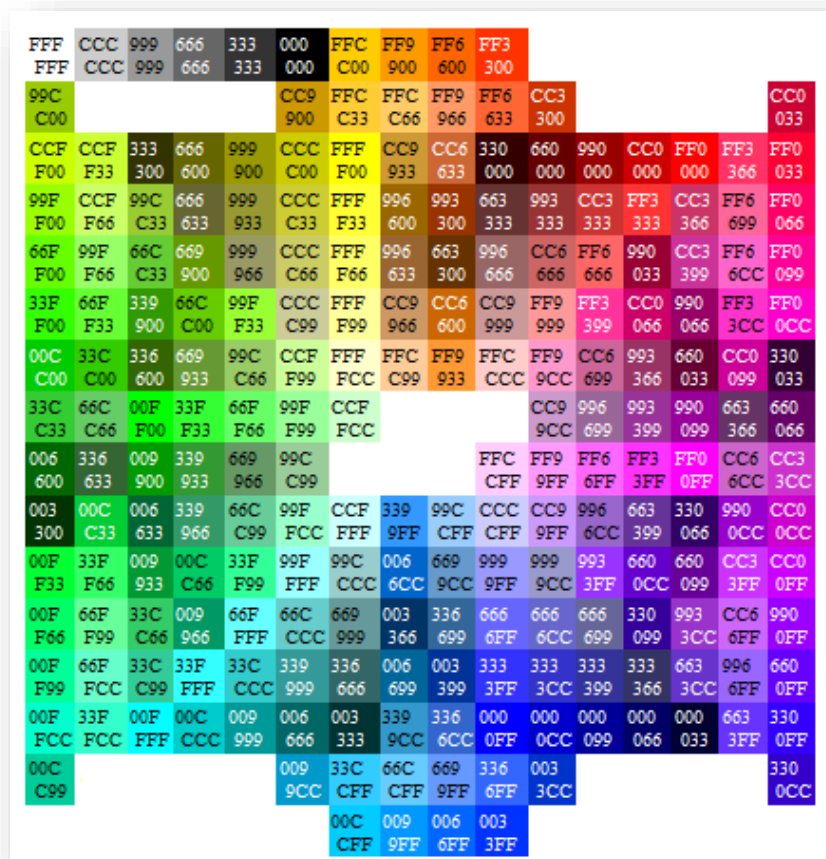
1.2.1. Форматування контейнером **style**

```
<!DOCTYPE html>
<html>

<head>
  <title>Контейнер style</title>
  <style type="text/css">
    body {background-color:#000}
    P {color:#FF3; text-align:right; font-style:italic; font-size:3rem; text-decoration: underline}
    H2 {color:#F03; text-align:center}
  </style>
</head>

<body>
  <H2>КПІ</H2>
  <P> КПІ </P>
</body>
</html>
```

* Розмір шрифту - властивість font-size. Для визначення розміру використовується одиниця **rem** (кореневий em / Root Em) - одиниця типографіки, що дорівнює кореневому (базовому) значенню font-size, тобто 1rem завжди дорівнюватиме значенню font-size, яке було визначено в html (як правило це 1rem = 16px). Якщо значення rem для <html> не задано, воно приймається за **1rem**, і всі інші розміри задаються щодо нього. Наприклад, **1.2rem** збільшить розмір шрифту на 20%.



1.2.2. Форматування атрибутом *style*

Синтаксис:

`<тег style="найменування властивості: значення">`

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title> Робота з font </title>
</head>

<body>
<H1>
  <P style="color:blue">Синій </P>
  <P style="color:red">Червоний</P>
  <P style="font-family:Arial">КПІ </P>
</H1>
</body>
</html>

```

1.2.3 Форматування правилом *@style*

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
<title>Правило @supports</title>

<style>
  P{
    width: 600px;
    padding: 1em;
    border: 1px solid blue;
  }

  @supports (color: navy){
    .green_text{color: green}
  }

  @supports not (color: superYellow){
    .orange_text{color: orange}
  }
</style>

</head>

<body>
  <P class="green_text">
```

```
    Текст 1 color: green!  
</P>  
  
    <P class="orange_text">  
        Текст 2 superYellow,  
        Текст 3 orange.  
    </P>  
</body>  
</html>
```

* [@supports](#) дозволяє перевірити браузерну підтримку якоїсь властивості або комбінації властивості та значення перед тим, як застосовувати певний блок стилів: CSS всередині буде застосований тільки якщо пара “властивість: значення” підтримується в конкретному браузері.

Якщо браузер користувача не підтримує CSS3, то відобразиться базовий вид елемента - це є так званий fallback (резервна) страховка.

2. Приклад створення однорівневого списку:

```
<!DOCTYPE html>  
<html>  
<head>  
    <title> Маркірований список </title>  
</head>  
  
<body>  
    <h2>Список рубрик</h2>  
<ul>  
    <li> Новини  
    <li> Фінанси  
    <li> Спорт
```

```
<li> Погода
</ul>
</body>
</html>
```

* Кожен список являє собою контейнер ``, який встановлює маркований список, або ``, який визначає список нумерований. Кожен елемент списку повинен починатися з тега ``

3. **Приклад вставки зображення:** * *name1.jpg* та *name2.jpg* – це реальні імена файлів-зображень, 1-ше зображення буде фоновим, а 2-ге – вставкою.

```
<!DOCTYPE html>
<html>
<head>
  <title> Зображення </title>
</head>
<body background="name1.jpg"> <body text="red">
  <center>
    <h2><marquee behavior=alternate direction="left"> Вставка зображення
  </marquee></h2>
    <br>
  </center>
</body>
</html>
```

4. **Приклад створення посилань в межах одного файлу:** * на місці має бути будь-який текст (текст можна скопіювати з іншого довільного текстового файлу .doc або .txt)

```
<!DOCTYPE html>
<html>
<head>
  <title> Посилання в межах одного файлу </title>
</head>
```

```
<body>
  <font size=4><b>Зміст</b></font><br>
  <a href="#глав1">Глава 1
</a><br>
  <a href="#глав2">Глава 2
</a><br>
  <a href="#глав3">Глава 3
</a><br>
```

```
<h1>Стаття</h1>
<h2 id="глав1">Глава 1</h2>
```

Текст 1 глави

.....
.....
.....
.....
.....
.....
.....

Будь-який текст

```
<h2 id="глав2">Глава 2 </h2>Глава 2 </h2>
```

Текст 2 глави

.....
.....

.....
.....
.....
.....

Будь-який текст

.....
.....

<h2 id="глав3">Глава 3 </h2>

Текст 3 глави

.....
.....

.....
.....
.....
.....
.....

Будь-який текст

</body>

</html>

5. Приклад створення посилань на інший файл: (в даному прикладі *name.html* – це ім'я файлу з попереднього (4-го) прикладу)

<!DOCTYPE html>

<html>

<head>

 <title> Посилання на інший файл </title>

</head>

<body>

```
<font size=4><b><u> Зміст</u> </b><font><br>
<a href="name.html#глав1">Глава1
</a><br>
<a href="name.html#глав2">Глава2
</a><br>
<a href="name.html#глав3">Глава3
</a><br>
</body>
</html>
```

6. Приклад реалізації посилання зображенням: * *name.jpg* – це реальне ім'я *jpg*-файлу, а *name.html* – будь-який *html*-файл.

```
<!DOCTYPE html>
<html>
<head>
  <title> Рисунок-посилання </title>
</head>
<body bgcolor="yellow"> <body text="Deepskyblue">
<center>
  <h2><marquee> Вставка зображення </marquee></h2>
  <a href="name.html" title="Посилання на файл"> </a><br>
</center>
</body>
</html>
```

Самостійне завдання

Створити **Головну сторінку**, фоном якої є будь-яка текстура, на сторінці розмістити нумерований список на кшталт такого:



Вибір **1 пункту** здійснити перехід на сторінку с Вашим фото і текстом (до якого застосовано обраний Вами стиль. Під ними розміщено слово НАЗАД (це повернення на Головну сторінку)

Вибір **2 пункту** переведе на сторінку с інформацією про групу, під якою розташована картинка, наприклад 🏠 (це повернення на Головну сторінку)

Вибір **3 пункту** переведе на будь-яку web-сторінку (наприклад, з прогнозом погоди)

Комп'ютерний практикум №2

Розмітка та компонування HTML-документа: Таблиці, Карта, Форми, Фрейми

Перша частина роботи складається з 2 тем, кожна з яких є окремою задачею і створюються окремим файлом в Блокноті (NotePad) або NotePad++

Файли зберігаються із розширенням *.html*

7. Приклад роботи з таблицями:

Для створення таблиць в HTML використовується елемент `<TABLE>`
....`</TABLE>`

Приклади фрагментів html-коду для створення таблиць:

- 1) без об'єднання комірок
- 2) з об'єднанням комірок


<pre><TABLE border=4 cellspacing=3> <CAPTION> Заголовок таблиці </CAPTION> <TR><TH bgcolor="white">Заголовок 1 <TH bgcolor="white">Заголовок 2 <TR><TD> Комірка 1 <TD> Комірка 2 <TR><TD> Комірка 3 <TD> Комірка 4 </TABLE></pre>	<table border="1"><thead><tr><th colspan="2">Заголовок таблиці</th></tr><tr><th>Заголовок 1</th><th>Заголовок 2</th></tr></thead><tbody><tr><td>Комірка 1</td><td>Комірка 2</td></tr><tr><td>Комірка 3</td><td>Комірка 4</td></tr></tbody></table>	Заголовок таблиці		Заголовок 1	Заголовок 2	Комірка 1	Комірка 2	Комірка 3	Комірка 4		
Заголовок таблиці											
Заголовок 1	Заголовок 2										
Комірка 1	Комірка 2										
Комірка 3	Комірка 4										
<pre><TABLE border=4 cellspacing=0 > <CAPTION>Заголовок таблиці </caption> <TR><TD bgcolor="white">Заголовок 1 <TD bgcolor="white">Заголовок 2 <TR><TD rowspan=3 bgcolor="white"> Комірка 1 <TD> Комірка 2 <TR><TD> Комірка 3 <TR><TD> Комірка 4 <TR><TD colspan=2 bgcolor="white" align="center"> Комірка 5 </table></pre>	<table border="1"><thead><tr><th colspan="2">Заголовок таблиці</th></tr><tr><th>Заголовок 1</th><th>Заголовок 2</th></tr></thead><tbody><tr><td rowspan="3">Комірка 1</td><td>Комірка 2</td></tr><tr><td>Комірка 3</td></tr><tr><td>Комірка 4</td></tr><tr><td colspan="2">Комірка 5</td></tr></tbody></table>	Заголовок таблиці		Заголовок 1	Заголовок 2	Комірка 1	Комірка 2	Комірка 3	Комірка 4	Комірка 5	
Заголовок таблиці											
Заголовок 1	Заголовок 2										
Комірка 1	Комірка 2										
	Комірка 3										
	Комірка 4										
Комірка 5											

Завдання:

Створити таблицю за наступним шаблоном

/ заголовок, текст в комірках і зображення, - за власним смаком ;) /

Освітні програми КПІ

Факультет	Спеціальність	
	Номер	Фото
ФММ	051 Економіка	
	051 Міжнародна економіка	
	051 Економічна кібернетика	
ФІМ	113 Прикладна математика	
	121 Інженерія програмного забезпечення	
	123 Комп'ютерна інженерія	

8. Створення карти <MAP>...</MAP>

Елемент **MAP** визначає набір областей графічного зображення, при якому різноманітні ділянки можуть посилатися на різні ресурси.

Малюнок з прив'язаними до нього активними областями називається **в картою-зображенням**.

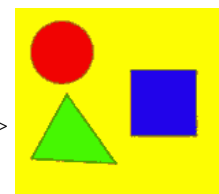
Така карта ззовні нічим не відрізняється від звичайного зображення, при цьому вона може бути розбита на невидимі області різної форми, при цьому кожен з таких контурів служить окремим посиланням.

Області задаються тегами **<AREA>...</AREA>** та атрибутом **shape**. Координати точок зображення можна визначити за допомогою будь-якого графічного редактора (наприклад **Paint**)

Приклад:

* файл **map.gif** може бути замінено на будь-яке зображення. Головне вказувати ім'я цього зображення в рамках тегу

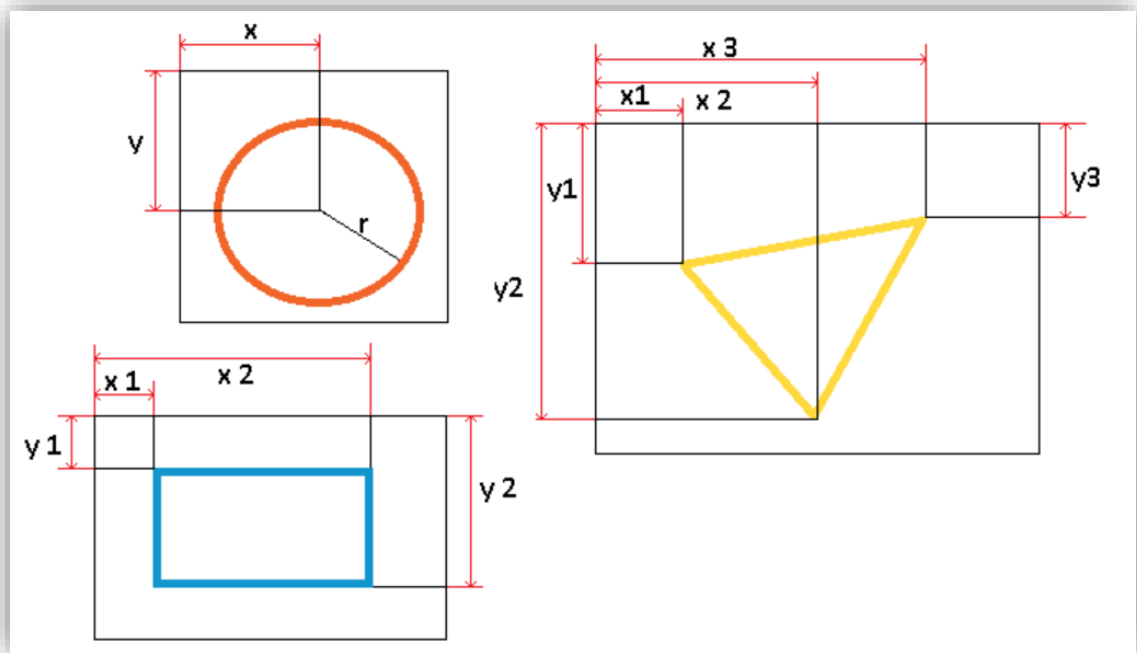
```
<IMG src="map.gif" usemap="#map.gif" border=0>
<MAP name="map.gif">
<AREA shape=circle coords="34,32,23" href="page1.html" alt="посилання 1">
<AREA shape=poly coords="12,110,37,62,72,114" href="page2.html" alt="посилання 2">
<AREA shape=rect coords="83,44,133,94" href="page3.html" alt="посилання 3">
</MAP>
```



Файл **map.gif**

Подивимось, як визначаються координати для фігур

!!! Зауважимо, що для $shape=poly$ кількість координат = кількості вершин багатокутника помножена на 2 (на кожну точку 2 координати x,y)



Завдання:

Створити карту зображення із використанням всіх видів контурів. Посилання виконуються на будь-які файли з першої лабораторної або зовнішні джерела.

«Форми, Фрейми»

9. Форми `<FORM>...</FORM>`

Форма HTML являє собою документ, створений з використанням елементів HTML. Призначенням форми є збір інформації від користувачів. Після того як користувач заповнить форму і запускає процес її обробки, інформація з неї потрапляє в програму, що працює на сервері. Інша програма під назвою **Common Gateway Interface** (CGI) обробляє її. Таким чином користувач може інтерактивно взаємодіяти з сервером Web через Internet. Форми так само зручні і для розробників сайту при розробці CMS (система управління вмістом (англ. **Content Management System**)), яка дозволяє підтримувати головну властивість сайту – актуальність

Форми – найважливіші інтерактивні елементи **HTML**. Весь опис HTML-форми розташовується усередині тегів **<form>...</form>**. Закриваючий тег **</form>** ніколи не несе в собі якої-небудь інформації, а от відкриваючий заслуговує додаткового розгляду. Розглянемо синтаксис цього тега:

<form method="метод" action="ім'я_сценарію">

Атрибут **Method** може приймати значення **GET** або **POST**. А атрибут **action** являє собою **URL** якого-небудь сценарію, що виконує дії відповідно до заповненої форми. (Головна відмінність методів **POST** і **GET** полягає у способі передачі інформації. У методі **GET** параметри передаються через адресний рядок, тобто по суті в **HTTP** заголовку запиту, в той час як у методі **POST** параметри передаються через тіло **HTTP**-запиту і ніяк не відбиваються на вигляді рядка)

GET Передача невеликих текстових даних на сервер; Пошук по сайту.

POST Пересилання файлів (фотографій, архівів, програм та ін.); відправлення коментарів; додавання та редагування повідомлень на форумі, блозі.

Ім'я_сценарію вказується в атрибуті **action** тега **<form>**

Наприклад:

action = "mailto:kpi@gmail.com"

action = "/example/forum.php"

action = "http://rozklad.kpi.ua "

Тег **<input>** має кілька атрибутів.

От основні з них: **<input type="тип поля" value="значення" name="ім'я">** Де **name** приймає будь-яке унікальне ім'я в межах однієї форми; будь-яке поле введення повинне мати ім'я, для наступного оброблення даних із цього поля сценарієм. **Value** - це значення за замовчуванням даного поля. Атрибут **type** може приймати значення **text**, **hidden**, **password**, **checkbox**, **radio**, **reset**, **submit**. Зупинимося на них.

Text - вказує браузеру, що потрібно відобразити поле введення. Значення атрибута **value** буде вказувати первісне значення даного поля. Також тут можуть бути присутнім два додаткових атрибуту, це **size** й **maxlength**. **Size** визначає довжину поля, а **maxlength** задає максимальну кількість символів, що може ввести користувач.

Hidden - цей тип текстового поля браузер на відображає. Він потрібний для приховання від відвідувача даних.

Password - цей тип поля відображається аналогічно текстовому, тільки всі введені в ньому символи будуть закриті зірочками, для того, щоб ніхто не міг підглянути пароль, що вводить вами.

Checkbox - браузер відображає цей тип поля квадратом, у якому можна поставити прапорець. За замовчуванням **value** даного поля коштує **ON**, але якщо поставити прапорець у це поле, то **value** прийме значення **checked**.

Radio - трохи схоже на **checkbox**, але можна вибрати тільки одне з декількох полів **radio**, у відмінності від **checkbox**, може бути включено.

Submit - відображає кнопку, при натисканні на яку всі дані з форми відправляються сценарію, зазначеному в атрибуті **action** тегу **<form>**.

Reset - при натисканні на цю кнопку, вся уведена інформація у форму анулюється.

Тег `<textarea>` визначає багаторядкову область введення. Цей тег має також кілька атрибутів. От приклад: `<textarea name="ім'я" rows="10" cols="20">Цей текст буде перебуває в області уведення, даного типу, користувач може змінити цей текст</textarea>` Атрибути `rows` й `cols` визначають висоту й довжину відповідно. "ім'я" - унікальне ім'я в межах однієї форми.

Теги `<select>...</select>`

Ці теги створюють список, кожний з варіантів якого задається тегом `<option>`. Тэг `<select>` може нести в собі атрибути `size`, `name` й `multiple`. З `size` й `name` все зрозуміло, а от з `multiple`... цей атрибут задає режим при якому користувач буде вибирати щось зі списку. Якщо цей атрибут присутній, то користувач може вибрати кілька значень зі списку, а якщо ж він опущений, то можна вибрати тільки одне значення.

Розглянемо приклад:

```
<html>
<head>
  <title>Анкета</title>
</head>
<body>
  <h3 align=left>Анкета</h3>
  <form>
    <b>Заповните наступні дані будь ласка:</b><br>
    Прізвище <input type=text size=14 name=fam><br>
    Ім'я <input type=text size=14 name=name><br>
    ВНЗ <input type=text size=25 value="КПІ ім. Ігоря Сікорського" name=vnz><br>
    <b>Скільки вам років?</b><br>
    <input name=1 type=radio value=1>Від 1 до 10<br>
    <input name=1 type=radio value =2>Від 10 до 18<br>
    <input name=1 type=radio value =3>Від 18 до 30<br>
    <input name=1 type=radio value =4>Від 30 до 50<br>
    <input name=1 type=radio value =5>Від 50 до 100<br>
  </form>
</body>
</html>
```

Завдання:

Створити форму за наступним шаблоном

/ текст та оформлення, - за власним смаком ;) /

10. Фрейми `<FRAMESET> ... </FRAMESET>`

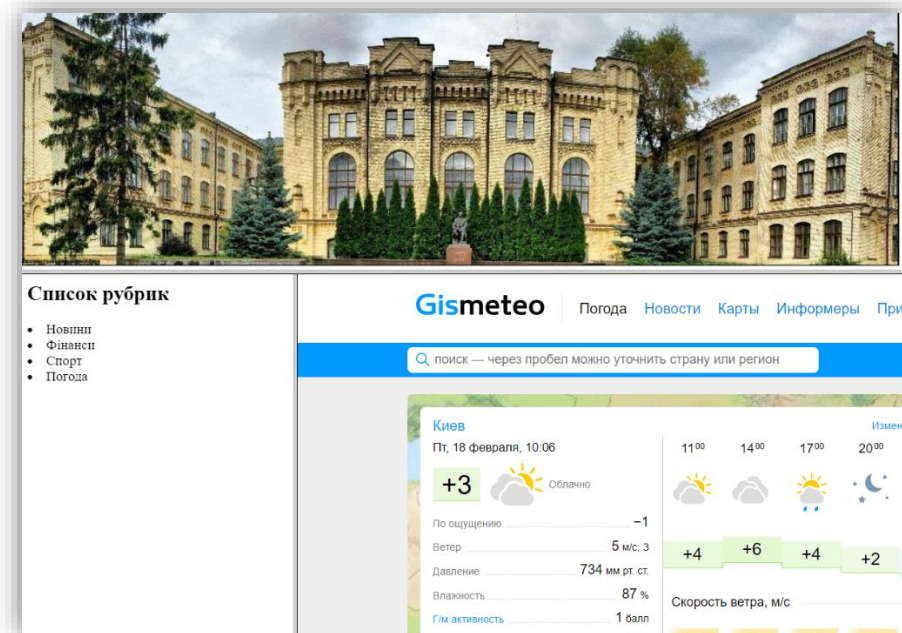
Використання `<FRAME>` дозволяє розміщувати у вікно робочої області одночасно декілька окремих HTML-документів, змінювати їх розміри, зміст, та організовувати зміну вмісту одного вікна після виконання користувачем дій

у іншому вікні. Саме ця властивість дозволяє використовувати фрейми у якості інструмента навігації.

Теги `<FRAMESET>` та `</FRAMESET>` замінюють теги `<BODY>` і `</BODY>` відповідно.

Всередині пари `<FRAMESET>... </FRAMESET>` можуть бути використані тільки теги `<FRAME>`, `<FRAMESET>` та `<NOFRAMES>`. Т.ч. створюється декілька окремих сторінок - *фреймів*, котрі виводяться на екран одночасно у вигляді декількох віконечок (згідно розбиттю на `rows` і `cols`) одного документа.

Приклад розбиття на фрейми:



```
<html>
<head>
  <title> Фрейми</title>
</head>
<frameset rows="295,*" >
  <frame src="КРІ.jpg" name="А">
  <frameset cols="20%,80%">
    <frame src="2.html" name="В">
    <frame src="https://www.gismeteo.ua/" name="С">
  </frameset>
</frameset>
</html>
```

* Зверніть увагу на атрибут **name** !!!

Наприклад в коді `<frame src="https://www.gismeteo.ua/" name="С">` - це означає, що області, в яку виводимо відомості про погоду ми надаємо ім'я "С". В майбутньому, якщо необхідно буде завантажити якийсь приклад саме в цю!!! область, то програмний код буде мати приблизно такий вигляд:

```
<a href="2.html" target="С">Приклад</a>
```

Тобто посилання знаходиться, в одному вікні, а файл `2.html` буде завантажено саме в область "С".

!!! Тег `<frameset>` більше НЕ підтримується в **HTML5**. Якщо ви хочете перевірити сторінку, яка містить фрейми, переконайтеся, що в `<!DOCTYPE>` встановлений як "HTML Frameset DTD" або "XHTML DTD Frameset"

В **HTML5** є аналог - тег `<iframe> ... </iframe>` (*inline frame - рядковий фрейм*) - тег-контейнер для автономної області на сторінці, в яку завантажуються самостійний документ, який визначається атрибутом `src`.

Завдання:

Створити фреймову структуру за наступним шаблоном

/ зміст та оформлення, - за власним смаком ;) /



1. При першому відкритті файлу в головній області має бути написано «Комп'ютерний Практикум», і це і буде вигляд головної (домашньої сторінки)
2. Список лівої області складається з трьох пунктів, кожен з яких посилається на: окремий файл (з 1 комп.практ.); будь-яке зображення та довільний сайт (зовнішнє посилання), при цьому файли завантажуються у Головну область

екрану! (* не всі сайти дозволяють інтегрувати себе у фрейми, тому виберіть той, хто дозволяє 😊)

3. Таблиця в правій частині має розмір 2 x 2 і містить невеликі зображення
4. Верхня (топ) область містить будь-яке зображення, клік на яке призведе до відображення Головної сторінки, тобто «Комп'ютерний Практикум» в головній області
5. Оформлення та контент – на власний смак 😊

Картинка-зображення		
Список	Головна область	Таблиця

Розмітка та компонування HTML-документа: Таблиці, Карта, Форми, Фрейми

Перша частина роботи складається з 2 тем, кожна з яких є окремою задачею і створюються окремим файлом в  Блокноті (NotePad) або  NotePad++

Файли зберігаються із розширенням *.html*

11. Приклад роботи з таблицями:

Для створення таблиць в HTML використовується елемент `<TABLE>`
....`</TABLE>`

Приклади фрагментів html-коду для створення таблиць:

- 3) без об'єднання комірок
- 4) з об'єднанням комірок

<pre><TABLE border=4 cellspacing=3> <CAPTION> Заголовок таблиці </CAPTION> <TR><TH bgcolor="white">Заголовок 1 <TH bgcolor="white">Заголовок 2 <TR><TD> Комірка 1 <TD> Комірка 2 <TR><TD> Комірка 3 <TD> Комірка 4 </TABLE></pre>	<table border="1"><thead><tr><th colspan="2">Заголовок таблиці</th></tr><tr><th>Заголовок 1</th><th>Заголовок 2</th></tr></thead><tbody><tr><td>Комірка 1</td><td>Комірка 2</td></tr><tr><td>Комірка 3</td><td>Комірка 4</td></tr></tbody></table>	Заголовок таблиці		Заголовок 1	Заголовок 2	Комірка 1	Комірка 2	Комірка 3	Комірка 4		
Заголовок таблиці											
Заголовок 1	Заголовок 2										
Комірка 1	Комірка 2										
Комірка 3	Комірка 4										
<pre><TABLE border=4 cellspacing=0 > <CAPTION>Заголовок таблиці </caption> <TR><TD bgcolor="white">Заголовок 1 <TD bgcolor="white">Заголовок 2 <TR><TD rowspan=3 bgcolor="white"> Комірка 1 <TD> Комірка 2 <TR><TD> Комірка 3 <TR><TD> Комірка 4 <TR><TD colspan=2 bgcolor="white" align="center"> Комірка 5 </table></pre>	<table border="1"><thead><tr><th colspan="2">Заголовок таблиці</th></tr><tr><th>Заголовок 1</th><th>Заголовок 2</th></tr></thead><tbody><tr><td rowspan="3">Комірка 1</td><td>Комірка 2</td></tr><tr><td>Комірка 3</td></tr><tr><td>Комірка 4</td></tr><tr><td colspan="2">Комірка 5</td></tr></tbody></table>	Заголовок таблиці		Заголовок 1	Заголовок 2	Комірка 1	Комірка 2	Комірка 3	Комірка 4	Комірка 5	
Заголовок таблиці											
Заголовок 1	Заголовок 2										
Комірка 1	Комірка 2										
	Комірка 3										
	Комірка 4										
Комірка 5											

Завдання:

Створити таблицю за наступним шаблоном

/ заголовок, текст в комірках і зображення, - за власним смаком ;) /

Освітні програми КПІ

Факультет	Спеціальність	
	Номер	Фото
ФММ	051 Економіка	
	051 Міжнародна економіка	
	051 Економічна кібернетика	
ФІМ	113 Прикладна математика	
	121 Інженерія програмного забезпечення	
	123 Комп'ютерна інженерія	

12. Створення карти <MAP>...</MAP>

Елемент **MAP** визначає набір областей графічного зображення, при якому різноманітні ділянки можуть посилатися на різні ресурси.

Малюнок з прив'язаними до нього активними областями називається **в картою-зображенням**.

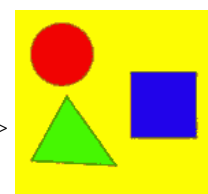
Така карта ззовні нічим не відрізняється від звичайного зображення, при цьому вона може бути розбита на невидимі області різної форми, при цьому кожен з таких контурів служить окремим посиланням.

Області задаються тегами **<AREA>...</AREA>** та атрибутом **shape**. Координати точок зображення можна визначити за допомогою будь-якого графічного редактора (наприклад **Paint**)

Приклад:

* файл **map.gif** може бути замінено на будь-яке зображення. Головне вказувати ім'я цього зображення в рамках тегу

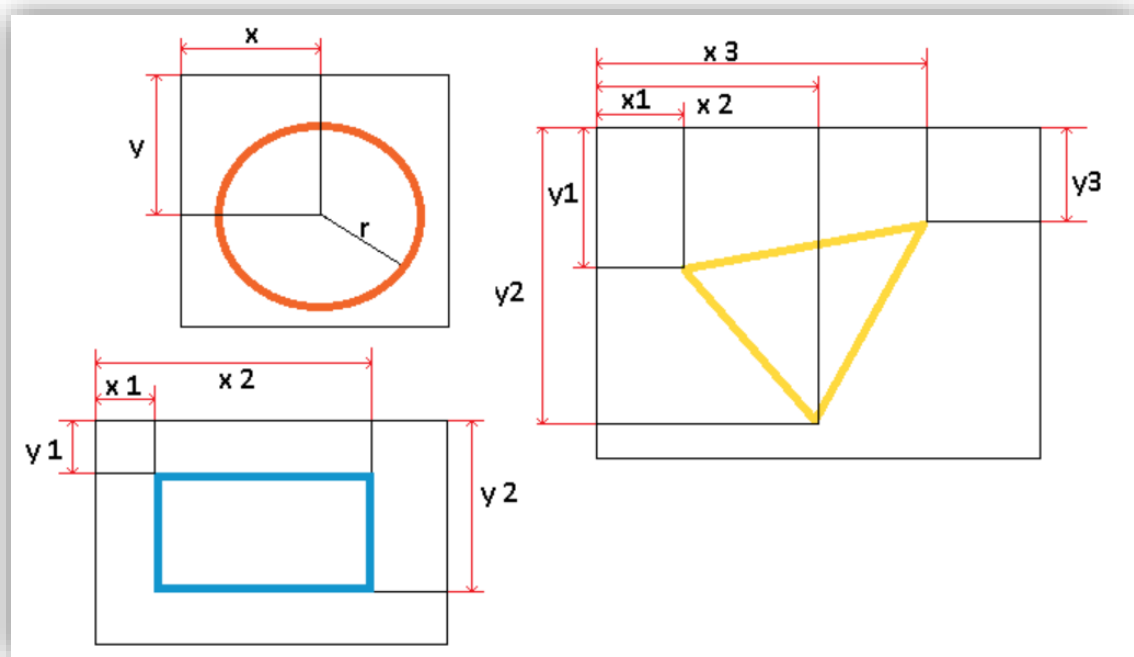
```
<IMG src="map.gif" usemap="#map.gif" border=0>
<MAP name="map.gif">
<AREA shape=circle coords="34,32,23" href="page1.html" alt="посилання 1">
<AREA shape=poly coords="12,110,37,62,72,114" href="page2.html" alt="посилання 2">
<AREA shape=rect coords="83,44,133,94" href="page3.html" alt="посилання 3">
</MAP>
```



Файл **map.gif**

Подивимось, як визначаються координати для фігур

!!! Зауважимо, що для $shape=poly$ кількість координат = кількості вершин багатокутника помножена на 2 (на кожену точку 2 координати x,y)



Завдання:

Створити карту зображення із використанням всіх видів контурів. Посилання виконуються на будь-які файли з першої лабораторної або зовнішні джерела.

«Форми, Фрейми»

13. Форми `<FORM>...</FORM>`

Форма HTML являє собою документ, створений з використанням елементів HTML. Призначенням форми є збір інформації від користувачів. Після того як користувач заповнить форму і запускає процес її обробки, інформація з неї потрапляє в програму, що працює на сервері. Інша програма під назвою **Common Gateway Interface** (CGI) обробляє її. Таким чином користувач може інтерактивно взаємодіяти з сервером Web через Internet. Форми так само зручні і для розробників сайту при розробці CMS (система управління вмістом (англ. **Content Management System**)), яка дозволяє підтримувати головну властивість сайту – актуальність

Форми – найважливіші інтерактивні елементи HTML. Весь опис HTML-форми розташовується усередині тегів `<form>...</form>`. Закриваючий тег

`</form>` ніколи не несе в собі якої-небудь інформації, а от відкриваючий заслуговує додаткового розгляду. Розглянемо синтаксис цього тега:

`<form method="метод" action="ім'я_сценарію">`

Атрибут `Method` може приймати значення `GET` або `POST`. А атрибут `action` являє собою URL якого-небудь сценарію, що виконує дії відповідно до заповненої форми. (Головна відмінність методів `POST` і `GET` полягає у способі передачі інформації. У методі `GET` параметри передаються через адресний рядок, тобто по суті в `HTTP` заголовку запиту, в той час як у методі `POST` параметри передаються через тіло `HTTP`-запиту і ніяк не відбиваються на вигляді рядка)

GET Передача невеликих текстових даних на сервер; Пошук по сайту.

POST Пересилання файлів (фотографій, архівів, програм та ін.); відправлення коментарів; додавання та редагування повідомлень на форумі, блозі.

Ім'я_сценарію вказується в атрибуті `action` тега `<form>`

Наприклад:

`action = "mailto:kpi@gmail.com"`

`action = "/example/forum.php"`

`action = "http://rozklad.kpi.ua"`

Тег `<input>` має кілька атрибутів.

От основні з них: `<input type="тип поля" value="значення" name="ім'я">` Де `name` приймає будь-яке унікальне ім'я в межах однієї форми; будь-яке поле введення повинне мати ім'я, для наступного оброблення даних із цього поля сценарієм. `Value` - це значення за замовчуванням даного поля. Атрибут `type` може приймати значення `text`, `hidden`, `password`, `checkbox`, `radio`, `reset`, `submit`. Зупинимося на них.

Text - вказує браузеру, що потрібно відобразити поле введення. Значення атрибута `value` буде вказувати первісне значення даного поля. Також тут можуть бути присутнім два додаткових атрибуту, це `size` й `maxlength`. `Size` визначає довжину поля, а `maxlength` задає максимальну кількість символів, що може ввести користувач.

Hidden - цей тип текстового поля браузер на відображає. Він потрібний для приховання від відвідувача даних.

Password - цей тип поля відображається аналогічно текстовому, тільки всі введені в ньому символи будуть закриті зірочками, для того, щоб ніхто не міг підглянути пароль, що вводить вами.

Checkbox - браузер відображає цей тип поля квадратом, у якому можна поставити прапорець. За замовчуванням `value` даного поля коштує `ON`, але якщо поставити прапорець у це поле, то `value` прийме значення `checked`.

Radio - трохи схоже на `checkbox`, але можна вибрати тільки одне з декількох полів `radio`, у відмінності від `checkbox`, може бути включено.

Submit - відображає кнопку, при натисканні на яку всі дані з форми відправляються сценарію, зазначеному в атрибуті `action` тегу `<form>`.

Reset - при натисканні на цю кнопку, вся уведена інформація у форму анулюється.

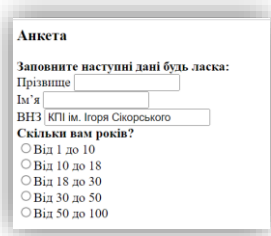
Тег `<textarea>` визначає багаторядкову область введення. Цей тег має також кілька атрибутів. От приклад: `<textarea name="ім'я" rows="10" cols="20">`Цей текст буде перебуває в області введення, даного типу, користувач може змінити цей текст`</textarea>`

Атрибути rows й cols визначають висоту й довжину відповідно. "ім'я" - унікальне ім'я в межах однієї форми.

Теги <select>...</select>

Ці теги створюють список, кожний з варіантів якого задається тегом <option>. Тэг <select> може нести в собі атрибути *size*, *name* й *multiple*. З *size* й *name* все зрозуміло, а от з *multiple*... цей атрибут задає режим при якому користувач буде вибирати щось зі списку. Якщо цей атрибут присутній, то користувач може вибрати кілька значень зі списку, а якщо ж він опущений, то можна вибрати тільки одне значення.

Розглянемо приклад:



```
<html>
<head>
<title>Анкета</title>
</head>
<body>
<h3 align=left>Анкета</h3>
<form>
<b>Заповните наступні дані будь ласка:</b><br>
Прізвище <input type=text size=14 name=fam><br>
Ім'я <input type=text size=14 name=name><br>
ВНЗ <input type=text size=25 value="КПІ ім. Ігоря Сікорського" name=vnz><br>
<b>Скільки вам років?</b><br>
<input name=1 type=radio value=1>Від 1 до 10<br>
<input name=1 type=radio value =2>Від 10 до 18<br>
<input name=1 type=radio value =3>Від 18 до 30<br>
<input name=1 type=radio value =4>Від 30 до 50<br>
<input name=1 type=radio value =5>Від 50 до 100<br>
</form>
</body>
</html>
```

Завдання:

Створити форму за наступним шаблоном

/ текст та оформлення, - за власним смаком ;) /



Успішність студентів

Прізвище:

Ім'я:

Група:

Навчання:

- Відмінно
- Добре
- Задовільно

Предмет:

- Математика
- Комп'ютерні технології
- Економіка

Додаткова інформація про успішність:

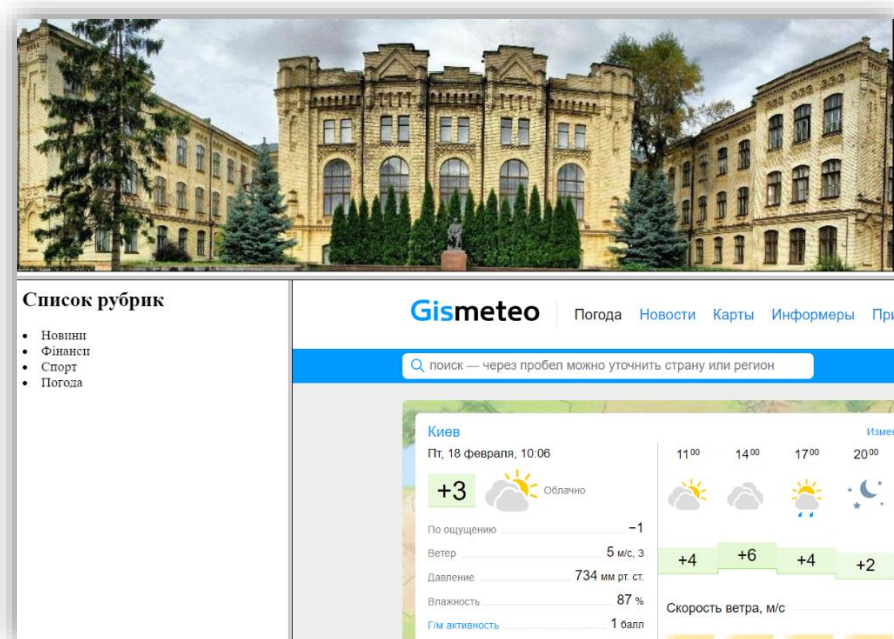
14. Фрейми <FRAMESET> ... </FRAMESET>

Використання <FRAME> дозволяє розміщувати у вікно робочої області одночасно декілька окремих HTML-документів, змінювати їх розміри, зміст, та організовувати зміну вмісту одного вікна після виконання користувачем дій у іншому вікні. Саме ця властивість дозволяє використовувати фрейми у якості інструмента навігації.

Теги `<FRAMESET>` та `</FRAMESET>` замінюють теги `<BODY>` і `</BODY>` відповідно.

Всередині пари `<FRAMESET>... </FRAMESET>` можуть бути використані тільки теги `<FRAME>`, `<FRAMESET>` та `<NOFRAMES>`. Т.ч. створюється декілька окремих сторінок - *фреймів*, котрі виводяться на екран одночасно у вигляді декількох віконечок (згідно розбиттю на *rows* і *cols*) одного документа.

Приклад розбиття на фрейми:



```
<html>
<head>
  <title> Фрейми</title>
</head>
<frameset rows="295,*" >
  <frame src="КРІ.jpg" name="А">
  <frameset cols="20%,80%">
    <frame src="2.html" name="В">
    <frame src="https://www.gismeteo.ua/" name="С">
  </frameset>
</frameset>
</html>
```

* Зверніть увагу на атрибут **name** !!!

Наприклад в коді `<frame src="https://www.gismeteo.ua/" name="С">` - це означає, що області, в яку виводимо відомості про погоду ми надаємо ім'я "С". В майбутньому, якщо необхідно буде завантажити якийсь приклад саме в цю!!! область, то програмний код буде мати приблизно такий вигляд:

```
<a href="2.html" target="С">Приклад</a>
```

Тобто посилання знаходиться, в одному вікні, а файл `2.html` буде завантажено саме в область "С".

!!! Тег `<frameset>` більше НЕ підтримується в **HTML5**. Якщо ви хочете перевірити сторінку, яка містить фрейми, переконайтеся, що в `<!DOCTYPE>` встановлений як "HTML Frameset DTD" або "XHTML DTD Frameset"

В **HTML5** є аналог - тег `<iframe> ... </iframe>` (*inline frame - рядковий фрейм*) - тег-контейнер для автономної області на сторінці, в яку завантажуються самостійний документ, який визначається атрибутом `src`.

Завдання:

Створити фреймову структуру за наступним шаблоном

/ зміст та оформлення, - за власним смаком ;) /

6. При першому відкритті файлу в головній області має бути написано «Комп'ютерний Практикум», і це і буде вигляд головної (домашньої сторінки)
7. Список лівої області складається з трьох пунктів, кожен з яких посилається на: окремий файл (з 1 комп.практ.); будь-яке зображення та довільний сайт (зовнішнє посилання), при цьому файли завантажуються у Головну область екрану! (* не всі сайти дозволяють інтегрувати себе у фрейми, тому виберіть той, хто дозволяє 😊)

8. Таблиця в правій частині має розмір 2 x 2 і містить невеликі зображення
9. Верхня (top) область містить будь-яке зображення, клік на яке призведе до відображення Головної сторінки, тобто «Комп'ютерний Практикум» в головній області
10. Оформлення та контент – на власний смак 😊

Картинка-зображення		
Список	Головна область	Таблиця

Комп'ютерний практикум №3

Робота в інтегрованому середовищі VB 6.0

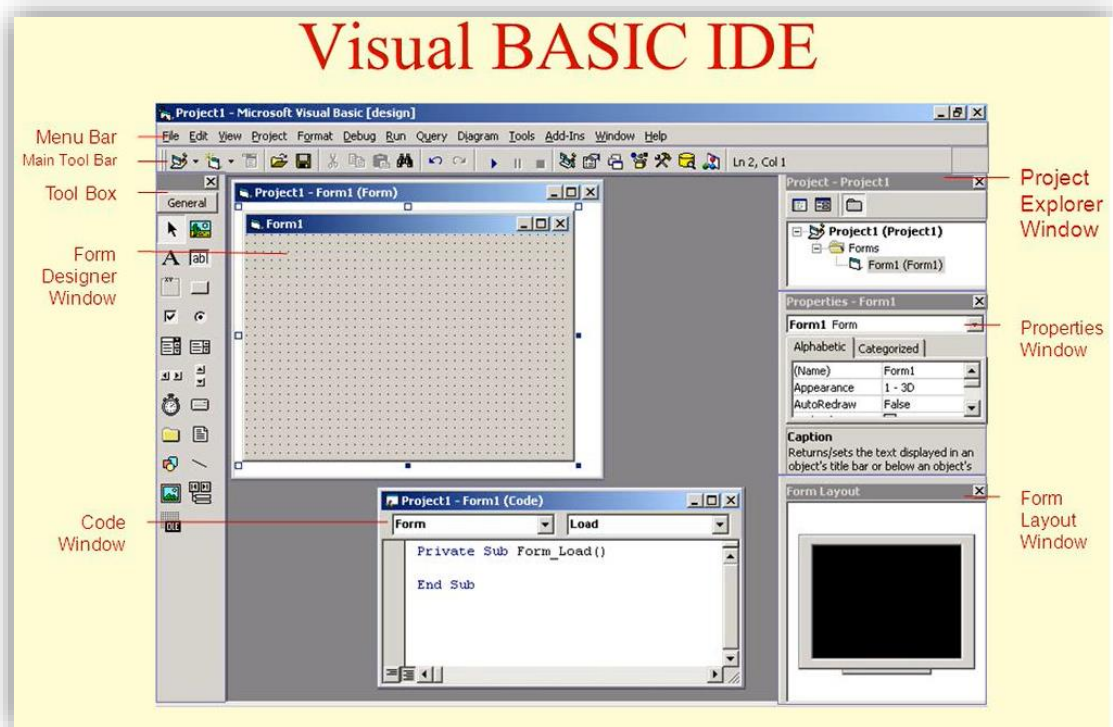
Основні етапи розробки програмних комплексів

Мета роботи:

- Ознайомитись з інтегрованим середовищем розробки додатків IDE MS VB;
- Вивчити основні команди створення, редагування, перегляду, збереження та запуску найпростіших програм на VB;
- Записати та виконати приклади програм;
- Створити програму власну програму за самостійним завданням

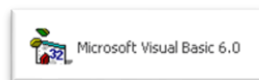
Знайомство с Visual Basic

1. Запустити Visual Basic



- *Перший рядок робочого екрана* - назва проекту та назва робочого програмного пакета;
- *Головне меню* містить команди для роботи над проектом в інтерактивному режимі;
- *Панель інструментів* знаходиться у верхній частині екрану. Її можна налагоджувати, так як це звичайно робиться у додатках Microsoft;

- *Панель елементів.* Елементи управління дозволяють користувачеві взаємодіяти з додатками. Панель елементів містить кнопки, поля введення, текстові наклейки та інші елементи, які необхідні для створення додатків. Щоб вибрати елемент управління потрібно встановити на нього вказівник миші і один раз або подвійно клацнути мишою;
- *Вікно проекту (Project Explorer).* У вікні проекту відображаються усі елементи програми: форми, модулі і т. ін., згруповані по категоріях. Усі додатки Visual Basic будуються за модульним принципом, а тому об'єктний код складається не із одного великого файлу, а із кількох частин. Кілька додатків можуть об'єднуватись у групи;
- *Файл проекту* – це повний список частин проекту. Проект складається із форм, модулів і файлів інших типів. Файл, що зберігає форму має тип *.FRM*, а файл модуля – тип *.BAS*. Якщо у формі використані графік або малюнок, або елемент керування зображенням, то Visual Basic створює файл із тим же іменем, але типу *.FRX*. Файли типу *.FRX* встановлюють зв'язок між файлами *.FRM* і графічними, вони подаються Visual Basic у двійковому форматі, а тому перейменовувати їх не рекомендується, тому що з часом знайти їх буде важко.




2. Після запуску VB на екрані з'явиться діалогове вікно, у якому можна вибрати тип додатку. Вибираємо Standard EXE.





Відкриється Головне вікно середовища VB, яке містить: панель інструментів; панель елементів; вікно форми; вікно властивостей; вікно коду.



У VB програма називається проект. Проект включає одну або декілька форм та код, що працює з даними формами.

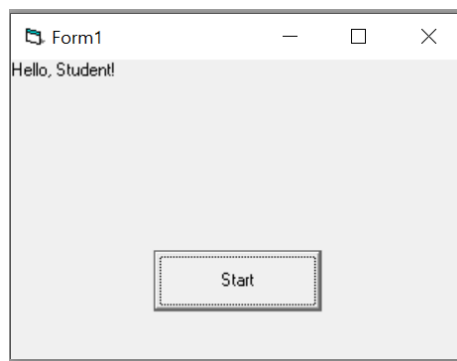
3. Створимо перший проект програми, що виводить повідомлення "Hello, Student!", після натиснення кнопки Start.

Спочатку розташуйте на формі елемент керування – кнопку  (Command). Це виконується подвійним кліком на відповідній піктограмі панелі елементів. На кнопці з'являється надпис (властивість Caption) Command1. Форма для

введення тексту у вікні  (Text), мітка-текст  (Label). Текст стандартного напису необхідно замінити на "Hello world". Для цього необхідно у вікні властивостей вибрати властивість Caption. Ввести текст, який, з'явиться на кнопці, а саме "Hello Student!". Подвійним клацанням по кнопці Command1 відкрийте вікно коду для операторів програми. Найпростіший оператор це *Print*, за допомогою якого можна виводити текст.

```
Private Sub Command1_Click()  
    Command1.Caption = "Start"  
    Print "Hello, Student!"  
End Sub
```

Щоб запустити програму на виконання, клацніть на кнопці Start , або натисніть клавішу [F5]. Завершити виконання програми можна за допомогою кнопки Stop  на панелі інструментів VB.



** Зверніть увагу, що Print завжди починає друкувати з верхнього лівого кута форми*

4. Створимо новий проект. На порожню форму поставимо кнопку. За допомогою **double click** на кнопці відкриємо вікно коду. Введемо наступну інструкцію, що здійснює доступ до властивостей об'єкта (*властивості елемента форми можна змінити як програмно, так і за допомогою вікна властивостей в самому інтегрованому середовищі*)

```
Private Sub Command1_Click()  
    Command1.Caption = "OK"  
End Sub
```

також можна, наприклад, змінити метод (виконати дію *Move*) і кнопка ОК буде переміщена у формі за відповідними координатами)

```
Private Sub Command1_Click()  
    Command1.Caption = "OK"
```

```
Command1.Move 200, 250
```

```
End Sub
```

Програма реалізації суми двох чисел виглядає наступним чином:

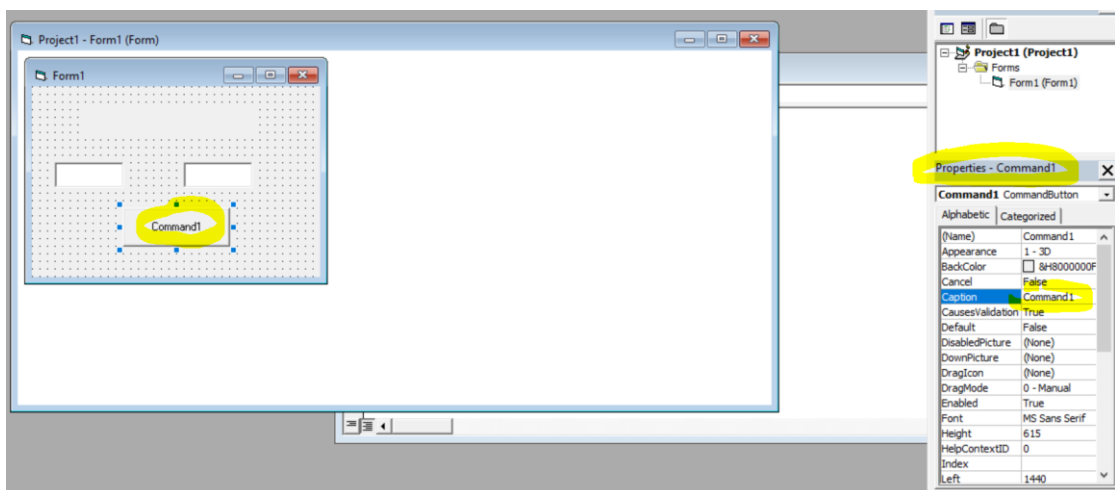
```
Private Sub Command1_Click()  
    Dim a As Integer  
    Dim b As Integer  
    Dim Sum As Integer  
    a = 12  
    b = 5  
    Sum = a + b  
    Print " Сума = "; Sum  
End Sub
```

** Всі недоліки, пов'язані з тим, що самі дані задаються в середині програми, а вивід здійснюється в ліву верхню зону форми, будуть усунені, коли будуть використані ввід через форму (елемент Text1) або InputBox та MsgBox*

Приклад: Введемо через форму два цілих числа **a** і **b**. Обчислити середнє арифметичне двох чисел. Результат вивести у окремому вікні - повідомленні

На порожню форму поставимо надпис, форму для вводу чисел (2 елемента Text), мітку для надпису «Введіть числа a та b.» (1 елемент Label), та кнопку (елемент Command) «ОК».

Для зручності всі властивості елементів форми будемо змінювати НЕ програмно, а у вікні **Properties**



```
Private Sub Command1_Click()
```

```
    Dim a As Integer
```

```
    Dim b As Integer
```

```
    Dim S As Double
```

```
    Label1.Caption = "Введіть числа a та b:"
```

```
    a = Val(Text1.Text)
```

```
b = Val(Text2.Text)
```

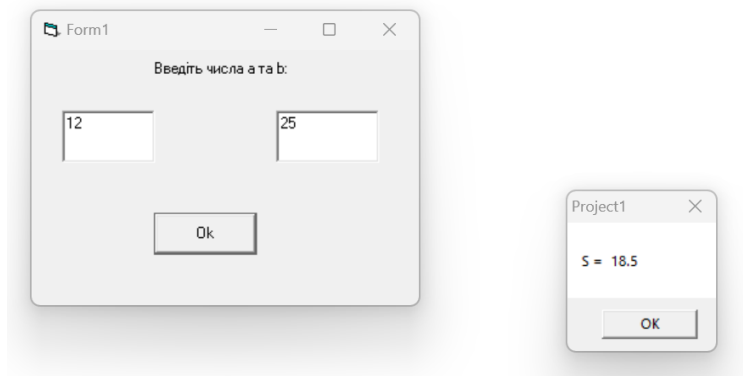
```
S = (a + b) / 2
```

```
MsgBox "S = " & Str(S)
```

```
End Sub
```

!!! Всі дані, що ми розміщуємо на форму або знімаємо з неї – це текстові дані. Щоб перевести текст в число і навпаки використовуємо функції **Val()** та **Str()** відповідно

Результат:



Самостійне Завдання:

1. Скласти програму, що обчислює суму, яку отримує вкладник банку під прості відсотки;
2. Скласти програму, що обчислює суму, яку отримує вкладник банку під складні відсотки;

$$F_i = P \left(1 + \frac{r}{100\%} \right)^n, \text{ де}$$

P - початковий внесок,
r - процентна ставка,
n - кількість років

* Користувач повинен мати можливість ввести суму вкладень, відсоток та термін вкладу.

** Обчислення відбуваються після натиснення відповідної кнопки «Прості» або «Складні»

*** Обидва завдання виконати на одній!!! формі.

Комп'ютерний практикум №4

Реалізація найпростіших програм в VB

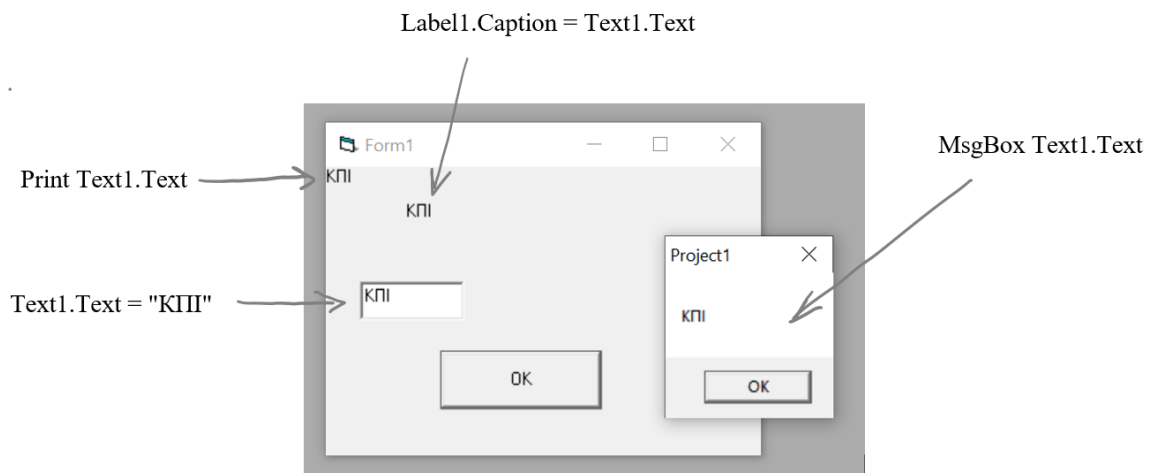
Мета роботи:

- навчитися використовувати та змінювати властивості елементів для організації діалогу користувача з VB;
- навчитися працювати зі змінними та константами, а також використовувати функції перетворення типів та математичні функції;
- створити власні програми за завданням.

Хід роботи:

На початку розглянемо різні варіанти введення/виведення даних в VB

```
Private Sub Command1_Click()  
    Text1.Text = "КПІ"  
    Label1.Caption = Text1.Text  
    Print Text1.Text  
    MsgBox Text1.Text  
End Sub
```



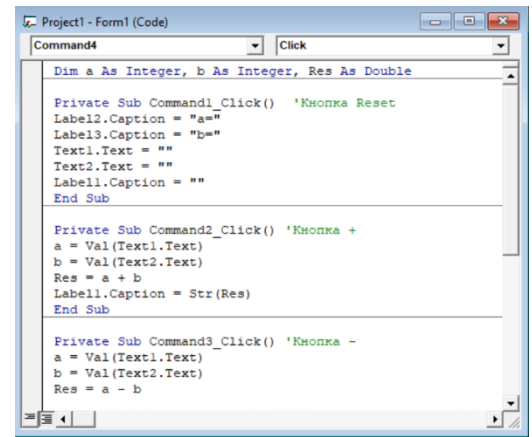
Приклад:

Створимо програму «Калькулятор». Програма буде реалізувати 4 основні дії: +, -, *, /

* Зауважимо, що кожна кнопка калькулятора створюється окремо на формі, після чого подвійним кліком викликається для редагування тіла для Sub.

Т.ч. програма буде складатися з п'яти підпрограм (під кожною кнопкою дії окремо та кнопку очистки Reset)

** Змінні *a*, *b*, *Res* мають бути оголошені ДО всіх підпрограм, щод кожен Sub мав до них доступ



```
Project1 - Form1 (Code)
Command4 Click
Dim a As Integer, b As Integer, Res As Double

Private Sub Command1_Click() 'Кнопка Reset
Label2.Caption = "a="
Label3.Caption = "b="
Text1.Text = ""
Text2.Text = ""
Label1.Caption = ""
End Sub

Private Sub Command2_Click() 'Кнопка +
a = Val(Text1.Text)
b = Val(Text2.Text)
Res = a + b
Label1.Caption = Str(Res)
End Sub

Private Sub Command3_Click() 'Кнопка -
a = Val(Text1.Text)
b = Val(Text2.Text)
Res = a - b
```

Код:

```
Dim a As Integer, b As Integer, Res As Double
```

```
Private Sub Command1_Click() 'Кнопка Reset
```

```
Label1.Caption = "a="
```

```
Label2.Caption = "b="
```

```
Text1.Text = ""
```

```
Text2.Text = ""
```

```
Label3.Caption = ""
```

```
End Sub
```

```
Private Sub Command2_Click() 'Кнопка +
```

```
a = Val(Text1.Text)
```

```
b = Val(Text2.Text)
```

```
Res = a + b
```

```
Label3.Caption = Str(Res)
```

```
End Sub
```

```
Private Sub Command3_Click() 'Кнопка -
```

```
a = Val(Text1.Text)
```

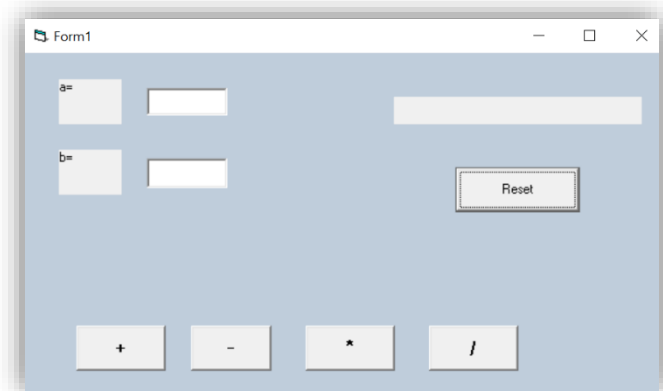
```
b = Val(Text2.Text)
```

```
Res = a - b
```

```
Label3.Caption = Str(Res)
```

```
End Sub
```

```
Private Sub Command4_Click() 'Кнопка *
```



```

a = Val(Text1.Text)
b = Val(Text2.Text)
Res = a * b
Label3.Caption = Str(Res)
End Sub
Private Sub Command5_Click()    ' Кнопка /
a = Val(Text1.Text)
b = Val(Text2.Text)
Res = a / b
Label3.Caption = Str(Res)
End Sub

```

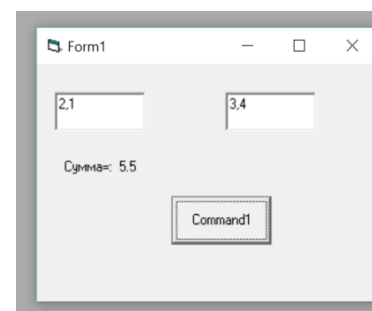
Приклад:

Робота з дійсними числами потребує іншого формату введення даних через елементи форми та використання функції переведення тексту в число *Cdbl()*

```

Private Sub Command1_Click()
    Dim Summ As Double
    Text1.Text = Replace(Text1.Text, ".", ",")
    Text2.Text = Replace(Text2.Text, ".", ",")
    Summ = Cdbl(Text2.Text) + Cdbl(Text1.Text)
    Label1.Caption = "Сума=" & Str(Summ)
End Sub

```



Математичні функції VB

Функція	Дія
Abs(n)	Повертає абсолютне значення n
Atn(n)	Повертає арктангенс n в радіанах
Cos(n)	Повертає значення косинуса n
Exp(n)	Повертає константу e в степені n
Rnd(n)	Генерує випадкове число між 0 і 1
Sin(n)	Повертає значення синуса n
Sqr(n)	Повертає квадратний корінь n
Str(n)	Перетворює числове значення в рядок
Tan(n)	Повертає значення тангенса n
Val(n)	Перетворює рядок в числове значення
Log(n)	Натуральний логарифм. Десятковий логарифм Log(n)/Log(10)

Завдання для самостійного розв'язку.

Варіант 1

1. Розмістити на формі два компонента: кнопку і Label. При затисненні на кнопку, на Label повинно з'явитися ваше прізвище.
2. Знайти значення параметра А рівняння $x^2+Ax+1=0$, якщо відомо, що коренем рівняння є число 1.
3. Знайти значення функції при $x = \pi/3$

$$y = \cos 3x - 2\sin\left(\frac{3\pi}{2} + x\right) - x^3$$

4. Знайти суму 20 перших членів геометричної прогресії, якщо перший член геометричної прогресії дорівнює 3, а знаменник геометричної прогресії дорівнює 2.

Варіант 2

1. Розмістити на формі два компонента: кнопку і Label. При затисненні на кнопку, на Label повинна з'явитися дата вашого народження.
2. Знайти значення параметра В рівняння $x^3+Bx^2+4x+5B=-1$, якщо відомо, що коренем рівняння є число 1.
3. Знайти значення функції при $x = 3\pi/4$

$$y = \operatorname{ctg}\left(\frac{3\pi}{2} - x\right) - \operatorname{ctg}^2 x + \frac{1+\cos 2x}{\sin^2 x}$$

4. Обчислити площу трикутника, якщо відомо сторони трикутника, що дорівнюють: 12;15;18 см.

Варіант 3

1. Розмістити на формі два компонента: кнопку і Label. При затисненні на кнопку, на Label повинна з'явитися назва вашої групи.
2. Знайти значення параметра С рівняння $3x^3+C^2x^2+4x+1=3$, якщо відомо, що коренем рівняння є число (-1)
3. Знайти значення функції при довільному x

$$y = 2^{2\operatorname{tg}|x|+\cos^2\sqrt{x}}$$

4. Скільки грошей отримає підприємець через рік, якщо він 250000грн покладе у банк під 8% річних .

Варіант 4

1. Розмістити на формі два компонента: кнопку і Label. При затисненні на кнопку, на Label повинна з'явитися назва міста, у якому ви народилися.
2. Обчислити площу трикутника АВС, заданого координатами його вершин (x_1 та y_1 для кожної точки).

3. Знайти значення функції при довільному x

$$y = \frac{\cos|2x|}{2\pi - x} - \sin^2\left(\frac{x - 3,1}{2\pi}\right).$$

4. Обчислити загальний прибуток від реалізації товарів, якщо відомо, що загальний обсяг виготовленої дорівнює Q , а ціна кожної одиниці товару дорівнює P .

Варіант 5

1. Розмістити на формі два компонента: кнопку і Label. При затисненні на кнопку, на Label повинно з'явитися прізвище вашого викладача.

2. Знайти значення параметра B рівняння $x^4 - \sqrt{B}x^3 + 4x^2 - 5\sqrt{B}x - 7 = 0$, якщо відомо, що коренем рівняння є число 1.

3. Знайти значення функції при $x=2$

$$y = \cos(x^2 + 1,43\pi) + \frac{x}{2}.$$

4. Валовий внутрішній продукт України дорівнює 179,7 млрд дол., а населення дорівнює 46 млн чоловік. Розрахуйте ВВП на душу населення в Україні.

Варіант 6

1. Розмістити на формі два компонента: кнопку і Label. При затисненні на кнопку, на Label повинно з'явитися ваше ім'я та по батькові.

2. Обчислити площу кола, якщо відомий радіус R .

3. Знайти значення функції при $x = 3\pi/4$

$$y = \operatorname{tg}\left(\frac{5\pi}{2} + x\right) - 3\operatorname{tg}^2x - (\cos 2x - 1) \cdot \cos^2x$$

4. Обчислити сторони трикутника ABC , заданого координатами його вершин (x_1 та y_1 для кожної точки).

Варіант 7

1. Розмістити на формі два компонента: кнопку і Label. При затисненні на кнопку, на Label повинна з'явитися назва вашої кафедри і спеціальності.

2. Знайти значення параметра A рівняння $x^4 + x^3 + Ax^2 + 1 = 0$, якщо відомо, що коренем рівняння є число -2 .

3. Знайти значення функції при $x = 0$

$$y = \sqrt{3} - \operatorname{tg}^2x - \operatorname{tg}\left(\frac{\pi}{3} - x\right)$$

4. Знайти суму 10 перших членів геометричної прогресії, якщо перший член геометричної прогресії дорівнює 6, а знаменник геометричної прогресії дорівнює 0,5.

Варіант 8

1. Розмістити на формі два компонента: кнопку і Label. При затисненні на кнопку, на Label повинна з'явитися назва вашої улюбленої дисципліни).
2. Знайти корені квадратного рівняння $ax^2 + bx + c = 0$ (a,b,c вводяться з форми)
3. Знайти значення функції при $x = \frac{\pi}{4}$

$$y = \sqrt{\cos^2 x + 0,5} + \sqrt{\sin^2 x + 0,5}$$

4. Знайти площу рівностороннього трикутника, сторона якого дорівнює 4см.

Варіант 9

1. Розмістити на формі два компонента: кнопку і Label. При затисненні на кнопку, на Label повинна з'явитися назва сьогоднішнього дня тижня.
2. Знайти значення параметра В рівняння $2x^4 - 5\sqrt{B}x^3 + 4x^2 + 10\sqrt{B} = 9$, якщо відомо, що коренем рівняння є число 1 (число задається з форми).
3. Знайти значення функції при $x = \frac{\pi}{3}$

$$y = 4\cos|x| - \sqrt{3}\operatorname{ctg}x - 1$$

4. Обчислити периметр прямокутника ABCD, заданого координатами його вершин.

Варіант 10

1. Розмістити на формі два компонента: кнопку і Label. При затисненні на кнопку, на Label повинна з'явитися назва даної дисципліни.
2. Знайти корінь рівняння $x^3 - 5x^2 - 16x - 4 = 0$.
3. Знайти значення функції при $x = \frac{\pi}{4}$

$$y = \frac{4\sin\left(\frac{\pi}{6}+x\right)\sin\left(\frac{5\pi}{6}+x\right)}{\cos^2 x} + 2|\operatorname{tg}x|$$

4. Обчислити площу трикутника, якщо відомо сторони трикутника, що дорівнюють: 6;8;10 см.

Комп'ютерний практикум №5

Програмування алгоритмів розгалужених обчислювальних процесів в реалізація найпростіших програм в VB

Мета роботи:

- вивчити методику програмування алгоритмів розгалужених обчислювальних процесів (*IF, SELECT CASE*)
- створити власні програми за завданням

Хід роботи:

1. Програма «Визначення max|min серед трьох чисел»

```
Private Sub Command1_Click()
```

*ці значення зручніше змінювати
безпосередньо у властивостях елементів*

```
Command1.Caption = "OK"
```

```
Label1.Caption = "a"
```

```
Label2.Caption = "b"
```

```
Label3.Caption = "c"
```

```
Label4.Caption = "max ="
```

```
Label5.Caption = "min ="
```

```
Label6.Caption = "Упорядкувати числа за спаданням"
```

```
a = Val(Text1.Text)
```

```
b = Val(Text2.Text)
```

```
c = Val(Text3.Text)
```

```
Text1.Text = ""
```

```
Text2.Text = ""
```

```
Text3.Text = ""
```

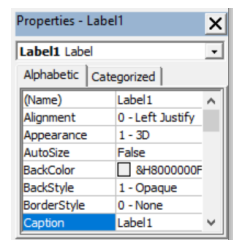
```
Max = a
```

```
If b > Max Then Max = b
```

```
If c > Max Then Max = c
```

```
Label7.Caption = Max
```

```
Min = a
```



*Приклад демонструє 2 окремих if, але
можна об'єднати умови за допомогою and*

```

If b < Min Then Min = b
If c < Min Then Min = c
Label8.Caption = Min
End Sub

```

2. Програма «Розрахунок податку від зарплати»

```

Private Sub Command1_Click()
    Command1.Caption = "підрахунок"
    Label1.Caption = "зарплата"
    Label2.Caption = "податок"
    Dim z As Integer, n As Integer
    z = Val(Text1.Text)
    If z < 200 Then
        n = z * 0.15
    ElseIf z >= 200 And z <= 500
    Then
        n = z * 0.2
    Else
        n = z * 0.28
    End If
    Label3.Caption = Str(n)
End Sub

```

3. Програма «Виведення тексту у відповідності до випадкового числа»

```

Private Sub Command1_Click()
    Command1.Caption = "Пуск"

```

Randomize

$n = \text{Int}(\text{Rnd} * 10) + 1$
чисел

'генератор випадкових

Label1.Caption = "Число - "

Label2.Caption = Str(n)

Label3.Caption = "Слово - "

Select Case n

Case 1

Label4.Caption = "Internet"

Case 2

Label4.Caption = "Україна"

Case 3

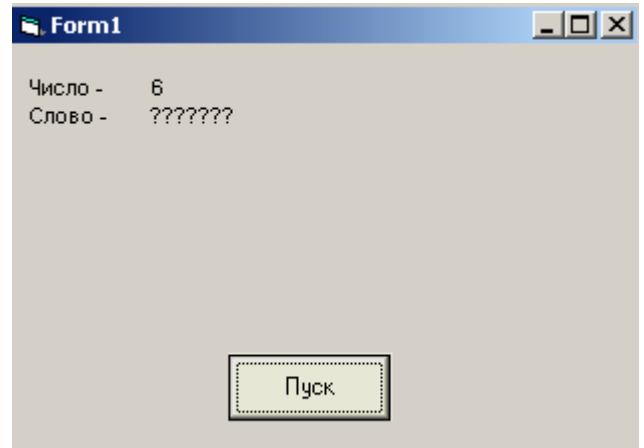
Label4.Caption = "Basic"

Case Else

Label4.Caption = "???????"

End Select

End Sub



Кожного разу, при натисненні на кнопку «Пуск», буде генеруватися випадкове число, відповідно – змінюватись слово.

Контрольні запитання.

1. Дія повного умовного оператора?
2. Що таке Блоковий синтаксис ?
3. Коли можна застосовувати скорочену форму умовного оператора?
4. Який синтаксис оператора Select Case?
5. Чим відрізняється оператор Select Case від умовного оператора?

Rnd()

Функція, що повертає **Single** , яке містить псевдовипадкове число.

Rnd [(*Number*)]

Необов'язковий аргумент *Number* є одиничним або будь-яким дійсним числовим виразом

Повертаються значення

Якщо число є	Rnd генерує
Менше нуля	Щоразу одне й те саме число, використовуючи число як початкове значення .
Більше за нуль	Наступне число в псевдовипадковій послідовності.
Дорівнює нулю	Останній згенерований номер.
Не надається	Наступне число в псевдовипадковій послідовності.

!!! Функція **Rnd** повертає значення менше 1, але \geq нулю

Значення *Number* визначає, як **Rnd** генерує псевдовипадкове число:

- Для будь-якого даного початкового насіння створюється та сама послідовність номерів, оскільки кожен наступний виклик функції **Rnd** використовує попереднє число як початкове число для наступного числа в послідовності.
- Перш ніж викликати **Rnd** , скористайтеся оператором **Randomize** без аргументу, щоб ініціалізувати генератор випадкових чисел за допомогою початкового значення на основі системного таймера.

Щоб отримати випадкові цілі числа в заданому діапазоні, використовуйте цю формулу:

```
Int((upperbound - lowerbound + 1) * Rnd + lowerbound)
```

Тут *upperbound*— це найвище число в діапазоні, а *lowerbound* — найменше число в діапазоні

* Щоб повторити послідовності випадкових чисел, викликайте **Rnd** з

негативним аргументом безпосередньо перед використанням **Randomize** з числовим аргументом. Використання **рандомізації** з тим самим значенням для числа не повторює попередню послідовність.

Наприклад, функція **Rnd** для створення випадкового цілого значення від 1 до 6.

```
Dim MyValue As Integer
MyValue = Int((6 * Rnd) + 1) ' Generate random value between 1 and 6.
```

Завдання для самостійного розв'язку

1. Використання оператора IF

1. Обчислити значення функції (a та x вводяться користувачем):

$$y = \begin{cases} \pi x^2 - 7/x^2 & \text{при } a < 1,3; \\ ax^3 + 7 \sqrt{\sin(ax)} & \text{при } a = 1,3; \\ \lg(ax + \sqrt{x}) & \text{при } a > 1,3. \end{cases}$$

2. Дано три відрізка A , B , C . Обчислити площу трикутника з сторонами A , B , C за формулою Герона, якщо його можна побудувати; якщо ні – видати відповідне повідомлення

3. Знайти квадрат найбільшого з трьох чисел A , B , C , і куб - найменшого з цих чисел

4. Визначити, чи значення змінних N і M кратні 3. Якщо обидва значення кратні 3, то обчислити їх суму, інакше обчислити їх різницю

5. Обчислити значення функції:

$$y = \begin{cases} \lg_3 a + \sqrt{x} / e^x & \text{при } x \leq 3; \\ \sqrt{ax} + 1/x & \text{при } 3 < x < 7; \\ \cos^2(ae^x) + a \sin^2(ae^x) & \text{при } x \geq 7. \end{cases}$$

6. Визначити мінімальний елемент із чотирьох X_1 , X_2 , X_3 , X_4 та його номер

7. Обчислити площі різних геометричних фігур і вивести на друк їх назву. Всі вхідні дані ($A B H R \varphi$) задаються на початку програми

$$\begin{cases} A \cdot B & \text{якщо } n = 1; \end{cases}$$

$$S = \begin{array}{ll} A \cdot H/2 & \text{якщо } n=2; \\ (A+B) \cdot H/2 & \text{якщо } n=3; \\ \pi R^2 & \text{якщо } n=4; \\ \pi R^2 \cdot \varphi/360 & \text{якщо } n=5. \end{array}$$

8. Ввести три цілих додатні числа A, B, C і число K (від 0 до 2). Обчислити значення функції:

$$y = \begin{cases} C^e + A/B & \text{при } K=0 \\ \ln(A+B+C) & \text{при } K=1 \\ \sqrt{(A-B)^{-C}} & \text{при } K=2 \end{cases}$$

9. Упорядкувати три числа X, Y, Z за збільшенням так, щоб змінній A відповідало найменше число, B - середнє, C - найбільше

2. Використання оператора SELECT

1. Ввести номер студента зі списку, вивести його прізвище
2. Вести номер місяця, вивести номер кварталу
3. Ввести телефонний код міста, вивести назву міста
4. Ввести номер автобуса, вивести кінцеву зупинку
5. Ввести першу букву назви річки, вивести її довжину

* Всі списки в завданні на SELECT складаються як мінімум з 5 елементів

** Завдання, що виділені маркером є обов'язковими до виконання! Решта, - за бажанням

Комп'ютерний практикум №6

Програмування алгоритмів циклічних обчислювальних процесів Робота з масивами

Мета роботи:

- Вивчити методику програмування алгоритмів циклічних обчислювальних процесів
- Реалізувати роботу з одновимірними масивами
- Створити власні програми за завданням

Хід роботи:

Оформлення роботи: Розмістити форму в центрі екрану, змінити розмір форми, кольори фону, символів, кнопки, - використовуючи панелі Form Layout та Properties Form (Alphabetic/Categorized)



Приклад: Програма визначення всіх тризначних цілих чисел, сума яких дорівнює 20

```
Private Sub Command1_Click()
```

```
    Command1.Caption = "OK"
```

```
    For i = 1 To 9
```

```
        For j = 1 To 9
```

```
            For k = 1 To 9
```

```
                s = i + j + k
```

```
                If s = 20 Then Print i; j; k
```

```
            Next k
```

```
        Next j
```

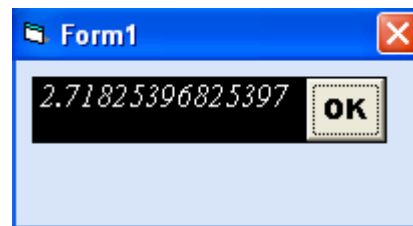
```
    Next i
```

End Sub



Приклад. Знайти суму членів ряду $S = 1 + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$. Обчислення проводити до тих пір, доки доданок не стане менше числа e (Зовнішній вигляд форми результату)

```
Private Sub Command1_Click()  
    Command1.Caption = "OK"  
    s = 1: x = 1  
    n = 1: y = 0.1  
    f = 1  
    e = 0.001      'задана точність  
    Do While y >= e  
        For i = 1 To n  
            f = f * i  
        Next i  
        y = x ^ n / f  
        s = s + y  
        n = n + 1  
        f = 1  
    Loop  
    Label1.Caption = Str(s)  
End Sub
```



Приклад. За запитом форми InputBox ввести 3 довільних слова, з яких формується одновимірний масив. Результат програми – вікно з переліком введених слів.

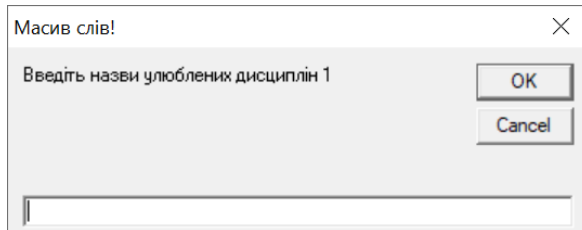
**Використовується чиста форма Form_Load*

```
Private Sub Form_Load()  
  
    Dim S1(1 To 3)  
  
    For i = 1 To 3  
  
        S1(i) = InputBox("Введіть назви улюблених дисциплін " & i, "Масив слів!")  
  
        Next i  
  
    str_msg = " "  
  
    For i = 1 To 3
```

```

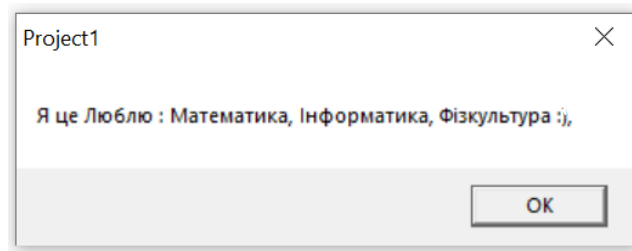
str_msg = str_msg & S1(i) & ", "
Next i
MsgBox "Я це Люблю :" & str_msg
End Sub

```



Вводимо назву...і так ще 2 рази...

Результат:



Приклад. Знаходження в одномірному масиві максимального та мінімального елементів, та їх порядкових номерів

```
Dim A(1 To 10) As Integer
```

```
Dim i As Integer, min As Integer, max As Integer, n As Integer
```

```
Dim S As String
```

```
Private Sub Command1_Click()
```

```
    Command1.Caption = "Розрахунок"
```

```
    Label1.Caption = " Програма знаходження min та max елементів масиву, та їх індексів."
```

```
    n1 = InputBox("Введіть кількість елементів <=10:")
```

```
    n = Val(n1)
```

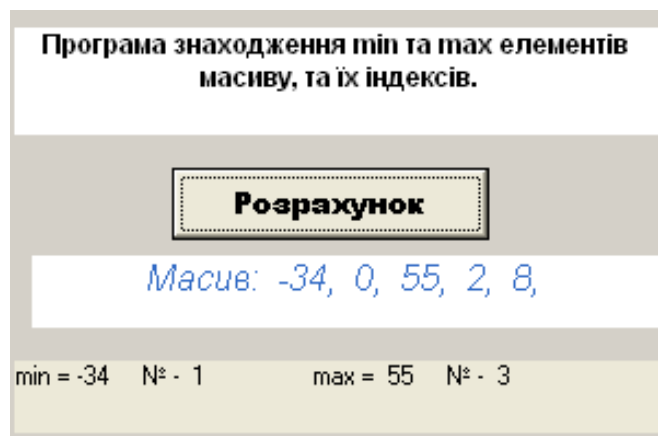
```
    For i = 1 To n
```

```

A(i) = InputBox("Введіть число:", "Масив")
Next i
min = A(1): max = A(1): kmin = 1: kmax = 1
For i = 2 To n
    If A(i) < min Then min = A(i): kmin = i Else

    If A(i) > max Then max = A(i): kmax = i Else
Next i
Label2.Caption = "min = " & Str(min) & " № - " & Str(kmin)
Label3.Caption = "max = " & Str(max) & " № - " & Str(kmax)
S = ""
For i = 1 To n
    S = S + Str(A(i)) & ", "
Next i
Label4.Caption = "Масив: " & S
End Sub

```



Завдання для самостійного розв'язку (згідно з варіантами)

**№ варіанту – порядковий номер студента в підгрупі*

** Розмірність матриці N задається користувачем з клавіатури.*

** При роботі з формами застосувати властивості змін зовнішнього вигляду.*

V.1. В одномірному масиві, який складається з N дійсних елементів, обчислити:

- 1) суму від'ємних елементів масиву;
- 2) добуток елементів масиву, що розташовані між мінімальним та максимальним елементами.

Впорядкувати елементи масиву за зростанням.

На екран вивести: вихідний масив; обчислені значення та перетворений масив.

V.2. В одномірному масиві, який складається з N дійсних елементів, обчислити:

- 1) суму додатних елементів масиву;
- 2) добуток елементів масиву, що розташовані між мінімальним та максимальним за модулями елементами.

Впорядкувати елементи масиву за спаданням.

На екран вивести: вихідний масив; обчислені значення та перетворений масив.

V.3. В одномірному масиві, який складається з N дійсних елементів, обчислити:

- 1) добуток елементів масиву, номери яких є парними;
- 2) суму елементів масиву, що розташовані між першим та останнім нульовими елементами.

Перетворити масив таким чином, щоб спочатку розміщувались всі додатні елементи, після чого – всі від'ємні (елементи, що дорівнюють 0 вважаються додатними).

На екран вивести: вихідний масив; обчислені значення та перетворений масив.

V.4. В одномірному масиві, який складається з N дійсних елементів, обчислити:

- 1) суму елементів масиву, номери яких є непарними;
- 2) суму елементів масиву, що розташовані між першим та останнім від'ємними елементами.

Стиснути масив, знищивши в ньому всі елементи, які менше за 1. Звільнені в кінці масиву елементи заповнити нулями.

На екран вивести: вихідний масив; обчислені значення та перетворений масив.

V.5. В одномірному масиві, який складається з N дійсних елементів, обчислити:

- 1) максимальний елемент масиву;
- 2) суму елементів масиву, що розташовані до останнього додатного елементу.

Стиснути масив, знищивши в ньому всі елементи, модуль яких знаходиться в інтервалі $[a, b]$. Звільнені в кінці масиву елементи заповнити нулями.

На екран вивести: вихідний масив; обчислені значення та перетворений масив.

V.6. В одномірному масиві, який складається з N дійсних елементів, обчислити:

- 1) мінімальний елемент масиву;
- 2) суму елементів масиву, що розташовані між першим та останнім додатними елементами.

Перетворити масив таким чином, щоб спочатку розміщувались всі елементи, які дорівнюють нулю, після чого – всі решта.

На екран вивести: вихідний масив; обчислені значення та перетворений масив.

V.7. В одномірному масиві, який складається з N дійсних елементів, обчислити:

- 1) номер максимального елемента масиву;
- 2) добуток елементів масиву, що розташовані між першим та другим нульовими елементами.

Перетворити масив таким чином, щоб в першій його половині розміщувались елементи, що розташовані на непарних позиціях, в другій половині – інші елементи.

На екран вивести: вихідний масив; обчислені значення та перетворений масив.

V.8. В одномірному масиві, який складається з N дійсних елементів, обчислити:

- 1) номер максимального елемента масиву;
- 2) суму елементів масиву, що розташовані між першим та другим від'ємними елементами.

Перетворити масив таким чином, щоб в першій його половині розміщувались елементи, модуль яких не перевищує 1., в другій половині – інші елементи.

На екран вивести: вихідний масив; обчислені значення та перетворений масив.

V.9. В одномірному масиві, який складається з N дійсних елементів, обчислити:

- 1) максимальний за модулем елемент масиву;
- 2) суму елементів масиву, що розташовані між першим та другим додатними елементами.

Перетворити масив таким чином, щоб всі елементи, які дорівнюють нулю розташовувались після всіх решта елементів.

На екран вивести: вихідний масив; обчислені значення та перетворений масив.

V.10. В одномірному масиві, який складається з N дійсних елементів, обчислити:

- 1) мінімальний за модулем елемент масиву;
- 2) суму елементів масиву, що розташовані після першого елемента, який дорівнює нулю.

Перетворити масив таким чином, щоб в першій його половині розміщувались елементи, що розташовані на парних позиціях, в другій половині – інші елементи.

На екран вивести: вихідний масив; обчислені значення та перетворений масив.

V.11. В одномірному масиві, який складається з N дійсних елементів, обчислити:

- 1) номер мінімального за модулем елемента масиву;
- 2) суму модулів елементів масиву, що розташовані після першого від'ємного елемента.

Стиснути масив, знищивши в ньому всі елементи, величина яких знаходиться в інтервалі $[a, b]$. Звільнені в кінці масиву елементи заповнити нулями.

На екран вивести: вихідний масив; обчислені значення та перетворений масив.

V.12. В одномірному масиві, який складається з N дійсних елементів, обчислити:

- 1) номер максимального за модулем елемента масиву;
- 2) суму елементів масиву, що розташовані після першого додатного елемента.

Перетворити масив таким чином, щоб спочатку розміщувались всі елементи, ціла частина яких знаходиться в інтервалі $[a, b]$, після чого – всі решта.

На екран вивести: вихідний масив; обчислені значення та перетворений масив.

V.13. В одномірному масиві, який складається з N дійсних елементів, обчислити:

- 1) кількість елементів масиву, що розташовані в діапазоні від A до B ;
- 2) суму елементів масиву, що розташовані після максимального елемента.

Впорядкувати елементи масиву за спаданням модулів елементів.

На екран вивести: вихідний масив; обчислені значення та перетворений масив.

V.14. В одномірному масиві, який складається з N дійсних елементів, обчислити:

- 1) кількість елементів масиву, що дорівнюють нулю;
- 2) суму елементів масиву, що розташовані після мінімального елемента.

Впорядкувати елементи масиву за зростанням модулів елементів.

На екран вивести: вихідний масив; обчислені значення та перетворений масив.

V.15 В одномірному масиві, який складається з N дійсних елементів, обчислити:

- 1) кількість елементів масиву, що більше за C ;
- 2) добуток елементів масиву, що розташовані після максимального за модулем елемента.

Перетворити масив таким чином, щоб спочатку розміщувались всі від'ємні елементи, після чого – всі додатні (елементи, що дорівнюють 0 вважаються додатними).

На екран вивести: вихідний масив; обчислені значення та перетворений масив.

V.16. В одномірному масиві, який складається з N дійсних елементів, обчислити:

- 1) кількість від'ємних елементів масиву;

2) суму елементів масиву, що розташовані після мінімального за модулем елемента.

Замінити всі від'ємні елементи масиву їх квадратами і впорядкувати елементи масиву за зростанням.

На екран вивести: вихідний масив; обчислені значення та перетворений масив.

V.17. В одномірному масиві, який складається з N дійсних елементів, обчислити:

- 1) кількість додатніх елементів масиву;
- 2) суму елементів масиву, що розташовані після останнього елемента, який дорівнює нулю.

Перетворити масив таким чином, щоб спочатку розміщувались всі елементи, ціла частина яких не перевищує 1, після чого – решта елементів.

На екран вивести: вихідний масив; обчислені значення та перетворений масив.

V.18. В одномірному масиві, який складається з N дійсних елементів, обчислити:

- 1) кількість елементів масиву, що менше за C ;
- 2) суму цілих частин елементів масиву, що розташовані після останнього від'ємного елемента.

Перетворити масив таким чином, щоб спочатку розміщувались всі елементи, які відрізняються від максимального не більш ніж на 20%, після чого – решта елементів.

На екран вивести: вихідний масив; обчислені значення та перетворений масив.

V.19. В одномірному масиві, який складається з N дійсних елементів, обчислити:

- 1) добуток від'ємних елементів масиву;
- 2) суму додатніх елементів масиву, що розташовані до максимального елемента.

Змінити порядок розташування елементів в масиві на зворотний.

На екран вивести: вихідний масив; обчислені значення та перетворений масив.

V.20. В одномірному масиві, який складається з N дійсних елементів, обчислити:

- 1) добуток додатних елементів масиву;
- 2) суму елементів масиву, що розташовані до мінімального елементу.

Впорядкувати за зростанням окремо елементи, що розташовані на парних місцях, і елементи, що розташовані на непарних місцях.

На екран вивести: вихідний масив; обчислені значення та перетворений масив.

Комп'ютерний практикум №7

Робота з двовимірними масивами

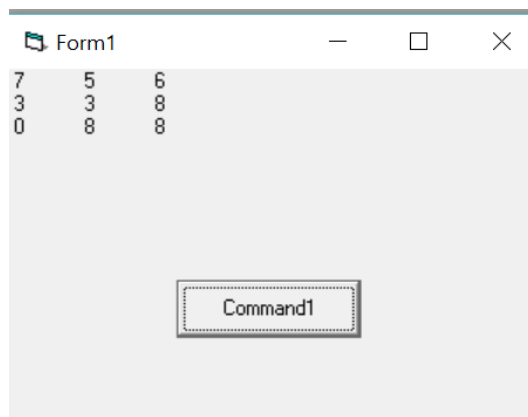
Мета роботи:

- Опрацювати циклічні алгоритмічні структури VB
- Реалізувати роботу з двовимірними масивами
- Створити власні програми за завданням.

(підготовка до роботи)

Приклад виводу масиву 3 x 3 із заповненням масиву випадковими числами
Rnd()

```
Private Sub Command1_Click()  
    Dim i As Integer, j As Integer  
    Dim A(1 To 3, 1 To 3) As Integer  
    For i = 1 To 3  
        For j = 1 To 3  
            A(i, j) = (Rnd * 8 + 1)  
            Print A(i, j); vbTab;  
        Next j  
    Print  
    Next i  
End Sub
```



Rnd

Синтаксис

```
Rnd[ (number) ]
```

Функція повертає значення типу **Single**, що містить випадкове число. Число, що повертається завжди менше 1, але більше 0. Для будь-якого заданого начального числа, генерується одна і та же сама послідовність, тому що кожен послідовний запит до функції **Rnd** використовує попередній номер як початкове число для наступного номера в послідовності.

Int((ВерхняГраниця - НижняГраниця + 1) * Rnd + НижняГраниця)"

Приклад: Знаходження мінімального елемента двовимірного масиву

```
Private Sub Command1_Click()
```

```
    Dim A() As Integer
```

```
    Dim m, n As Integer
```

```
    Dim i, j, min As Integer
```

```
    Dim S As String
```

```
    n = Val(Text1.Text)
```

```
    m = Val(Text2.Text)
```

```
    ReDim A(1 To n, 1 To m)
```

```
    S=""
```

```
    For i = 1 To n
```

```
        For j = 1 To m
```

```
            A(i, j) = Int(Rnd * 10 + 1)
```

```
            S = S & A(i, j) & " "
```

```
        Next j
```

```
        S = S & vbCrLf
```

```
    Next i
```

```
Label1.Caption = S
```

```
min = A(1, 1)
```

```
For i = 1 To n
```

```
    For j = 1 To m
```

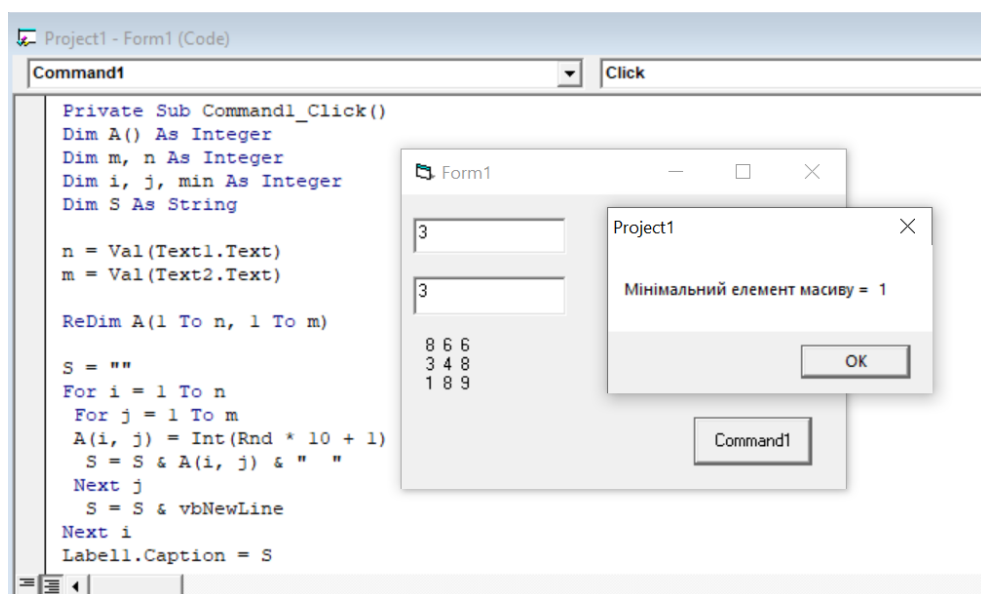
```
        If A(i, j) < min Then min = A(i, j)
```

```
    Next j
```

```
Next i
```

```
MsgBox "Мінімальний елемент масиву = " & Str(min)
```

```
End Sub
```



Завдання для самостійного розв'язку

В.1. Дано дві двомірні прямокутні матриці А і В. Поміняти місцями рядок матриці А, сума елементів якого найменша, зі стовпчиком матриці В, сума елементів якого найбільша. На екран монітору вивести і вихідні, і перетворені матриці.

В.2. Відсортувати двомірну прямокутну матрицю цілих чисел А:

- 1) по рядках, у порядку зростання елементів в рядку;
- 2) по стовпчикам, у порядку спадання елементів у стовпчику.

На екран монітору вивести і вихідну, і перетворені матриці.

V.3. Перемножити, якщо це можливо, дві двомірні прямокутні матриці A і B . В іншому випадку - вивести на екран повідомлення про несумісність матриць. На екран монітору вивести і вихідні матриці, і їх добуток.

V.4. З двох двомірних матриць A і B створити матрицю C за наступним правилом: якщо сума індексів відповідного елемента масиву C парна, то елемент переноситься із масиву A , в протилежному випадку - із масиву B . На екран монітору вивести і вихідні, і створену матриці.

V.5. Дано дві двомірні матриці A і B . Якщо матриця A симетрична відносно головної діагоналі, а матриця B симетрична відносно побічної діагоналі, то:

- 1) в матриці A поміняти місцями мінімальне і максимальне значення;
- 2) в матриці B всі числа, що кратні 5, замінити на 0.

На екран монітору вивести і вихідні, і перетворені матриці.

V.6. Відсортувати двомірний масив A в порядку зростання елементів у напрямку руху по спіралі за годинниковою стрілкою від елемента $A[1,1]$. На екран монітору вивести і вихідну, і перетворену матрицю.

V.7. Дано двомірну прямокутну матрицю A . Переставляючи її стовпчики та рядки, досягти того, щоб найбільший елемент (або один з них, якщо їх декілька) опинився у верхньому лівому куті. На екран монітору вивести і вихідну, і перетворену матриці.

V.8. Дано двомірний масив A ($2 \times N$). Розглядаючи цей масив, як координати N точок, знайти найбільшу відстань між точками. На екран монітору вивести і вихідну матрицю, і максимальну відстань, і № точок, для яких ця відстань отримана.

V.9. Елемент матриці назвемо "сідловим", якщо він найменший у рядку і одночасно найбільший у стовпчику. Для заданої двомірної прямокутної матриці визначити кількість "сідлових" елементів. На екран монітору вивести вихідну матрицю, кількість "сідлових" елементів, їх індекси.

V.10. Дана двомірна матриця дійсних чисел A , всі елементи якої різні. Знайти скалярний добуток рядку, в якому знаходиться найбільший елемент матриці, на стовпчик з найменшим елементом. На екран монітору вивести і вихідну матрицю, і визначений добуток.

V.11. Дано дві двомірні матриці A і B . Якщо матриця A містить тільки додатні елементи, а матриця B симетрична відносно головної діагоналі, то:

- 1) в матриці B всі числа, що кратні 3, замінити на 1.
- 2) в матриці A поміняти місцями стовпчики, що містять мінімальне і максимальне значення;

На екран монітору вивести і вихідні, і перетворені матриці.

V.12. З двох двомірних матриць A і B створити матрицю C за наступним правилом: якщо сума індексів відповідного елемента масиву C непарна, то елемент переноситься із масиву A , в протилежному випадку - із масиву B . Після формування масиву C всі додатні числа збільшити на 1. На екран монітору вивести і вихідні, і створену матриці.

V.13. Відсортувати двомірний масив A в порядку спадання елементів у напрямку руху по спіралі проти годинниковою стрілкою від елемента $A[1,1]$. На екран монітору вивести і вихідну, і перетворену матрицю.

V.14. Дано дві двомірні прямокутні матриці A і B . Поміняти місцями рядок матриці A , який містить найбільшу кількість додатних елементів, зі стовпчиком матриці B , який містить найменшу кількість від'ємних елементів. На екран монітору вивести і вихідні, і перетворені матриці.

V.15 Відсортувати двомірну прямокутну матрицю цілих чисел A :

- 1) по рядках, у порядку спадання елементів в рядку;
- 2) по стовпчикам, у порядку зростання елементів у стовпчику.

На екран монітору вивести і вихідну, і перетворені матриці.

V.16. Дано дві двомірні прямокутні матриці дійсних чисел A і B , всі елементи яких є різними. Знайти скалярний добуток стовпчику матриці A , в якому знаходиться найбільший елемент матриці, на рядок матриці B , що містить найменший елемент. На екран монітору вивести: вихідні матриці, визначений добуток або повідомлення, у разі, якщо множення недопустиме.

V.17. Дано дві двомірні прямокутні матриці цілих чисел A і B . Якщо суми рядків матриці A співпадають з відповідними сумами рядків матриці B , то поміняти місцями елементи головних діагоналей матриць. На екран монітору вивести і вихідні, і перетворені матриці.

V.18. Дано двомірну прямокутну матрицю A . Переставляючи її стовпчики та рядки, досягти того, щоб у верхньому лівому куті опинився елемент (або один з них, якщо їх декілька), який містить найбільшу кількість нулів. На екран монітору вивести і вихідну, і перетворену матриці.

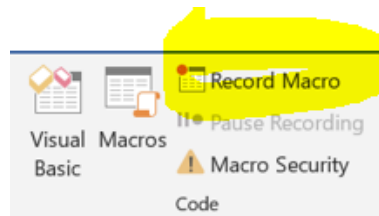
Комп'ютерний практикум №8

Створення макросів у додатку MS Word (макрорекордер та код на VBA)

У програмі Word можна автоматизувати часто виконувані завдання, створюючи та запускаючи макроси. **Макрос** – це набір команд та інструкцій, об'єднаних в одну команду для автоматичного виконання завдання.

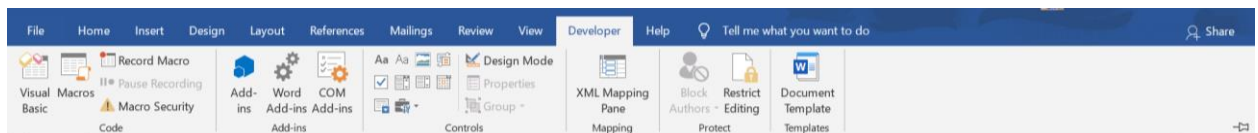
Пакет Word у поєднанні з VBA забезпечує широкі можливості для роботи з текстом або фрагментами тексту.

Створювати макроси можна *вручну* або *автоматично*. Останній варіант передбачає запис дій, які ми виконуємо в програмі, для їх подальшого повтору. Це досить простий спосіб, користувач не повинен мати якісь навички кодування тощо. Однак, саме в зв'язку з цим, застосувати його можна не завжди і до досить обмеженого спектру завдань.

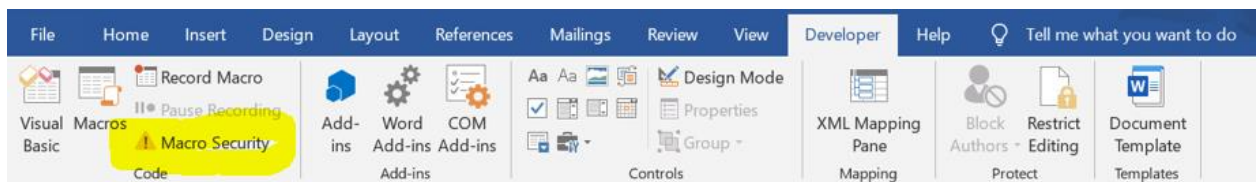


Щоб створювати макроси вручну, потрібно вміти програмувати. Але саме такий спосіб іноді є єдиним або одним з небагатьох варіантів ефективного вирішення поставленого завдання.

На початку роботи в офісному додатку необхідно включити вкладку/*Ribbon Developer tab «Developer»* (Option – Customize Ribbon – Delelop)



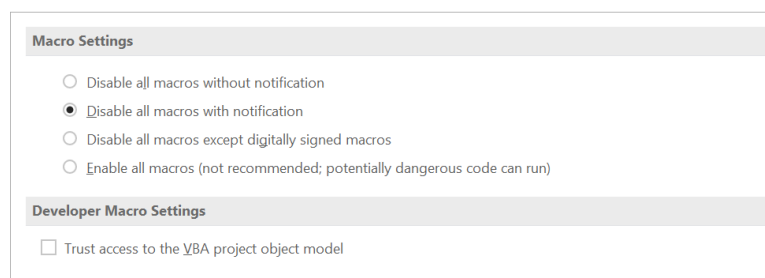
Після того, як *Developer* буде активовано, необхідно включити макроси. На вкладці *Developer* знаходимо опцію щодо **Безпеки макросів**



У вікні з налаштуваннями можна активувати відразу все макроси. Однак, слід звернути увагу на те, що розробники Microsoft не рекомендують вибирати цей

варіант, так як є ймовірність запуску небезпечної програми, що може нашкодити комп'ютеру. Тому, виконуючи цю операцію, пам'ятайте, що ви дієте на свій страх і ризик. Рекомендується 2-й пункт ☺

1. **Вимкнути всі макроси без сповіщення:** *вимикає всі макроси в документах та всі попередження про безпеку, пов'язані з макросами. Ця опція полягає у виборі, чи не довіряєте ніякому макросу.*
2. **Вимкнути всі макроси за допомогою сповіщення:** *це налаштування за замовчуванням. Це дозволяє активувати макроси у кожному конкретному випадку. Макроси вимкнені, але попередження про безпеку з'являються у присутності макросу.*
3. **Вимкнути всі макроси, за виключенням цифрових підписаних макросів:** *так само, як і в налаштуваннях за умовчанням, ця опція все ще дозволяє цифровим підписаним макросам, надісланим електронною поштою для запуску. Якщо це не так, ви отримаєте сповіщення, що дасть вам можливість це зробити;*
4. **Увімкнути всі макроси:** *цей параметр сильно не рекомендується, оскільки він показує комп'ютер, що виконує шкідливий код.*



Далі можна розпочинати роботу з макросами.

Для доступу до тих чи інших фрагментів документа використовують такі об'єкти як **Sections** (розділи), **Paragraphs** (абзаци), **Sentences** (речення), **Words** (слова) та ін.

Наприклад:

<i>Words(6)</i>	<i>дає можливість перейти до шостого слова у документі</i>
<i>Sentences(11)</i>	<i>перехід до одинадцятого речення</i>

Після переходу до певної частини документа можна використовувати об'єкт **Font**, який дозволяє змінювати властивості шрифту. Цей об'єкт містить методи в панелі меню «Формат | Шрифт»

Об'єкт **Range** є базовою в операції редагування. Його можна розглядати як безперервний фрагмент тексту (не пов'язаний з виділеною областю), який залежить від положення курсору. Range включає такі об'єкти, як **Words** (слова) або **Sentences** (речення); при зверненні до них сам об'єкт Range не потрібно вказувати, тоді як для Sections (розділи) або Paragraphs (абзаци) це робити необхідно.

Об'єкт **Range** підтримує основні методи, якими користуються при форматуванні тексту (такі як «Копіювати», «Вставити», «Вирізати», команди меню «Формат»).

Наприклад, для пошуку та заміни тексту в макросі використовуються наступні рядки

```
.Text = "."
.Replacement.Text = ""
```

Найбільш популярні заміни:

Що змінюємо	Знайти	Замінити на
прибрати прогалини між словом та точкою	"([а-я])@(\.)"	"\1\2"
додати пробіл між точкою та наступною пропозицією	"([а-я].@)([А-Я])"	"\1 \2"
прибрати прогалини між словом та комою	"([а-я])@(\,)"	"\1\2"
додати пробіл між комою та наступним словом	"([а-я],)([а-я])"	"\1 \2"
прибрати пробіли між точкою та переходом на наступний рядок	"(\.)(^13)"	"\1\2"
прибрати повторювані прогалини	"(\s){2;}"	" "
прибрати пробіл після дужки, що відкриває	"\(" "	"(" "
прибрати пробіл перед закриває дужки	"\)" "	")"

Приклади:

- змінює колір тексту в першому абзаці в активному документі
`ActiveDocument.Paragraphs(1).Range.Font.ColorIndex = wdGreen`
- форматує виділений текст, щоб він відображався червоним кольором.
`Selection.Font.ColorIndex = wdRed`

Приклади коду (*Primer* – назва документу, може бути змінена на власну)

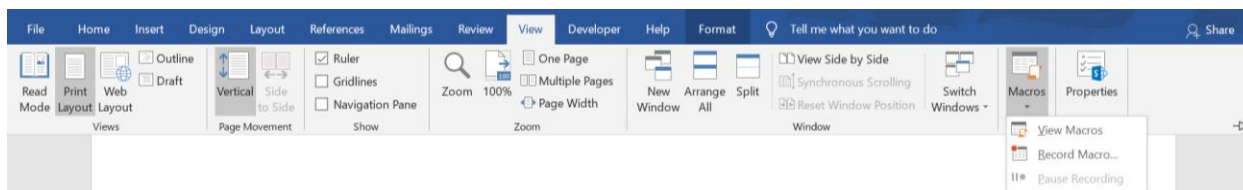
<code>Primer.Sentences(2).Font.ColorIndex = wdGreen</code>	<i>В 2 реченні встановлюється зелений колір шрифту</i>
<code>Primer.Sentences(2).Font.Size = 16</code>	<i>В 2 реченні встановлюється кегль 16 пт</i>
<code>Primer.Sentences(2).Font.Underline = wdUnderlineDouble</code>	<i>В 2 реченні встановлюється шрифт з подвійним підкресленням</i>
<code>Primer.Sentences(2).Font.Name = "Arial"</code>	<i>В 2 реченні встановлюється гарнітура Arial</i>

<code>Primer.Range(1,15).Bold = True</code>	<i>Символи з 1 по 15 виділяються жирним шрифтом</i>
<code>Primer.Paragraphs(5).Range.InsertAfter «КПІ»</code>	<i>Після 5 абзацу вставляється текст КПІ</i>

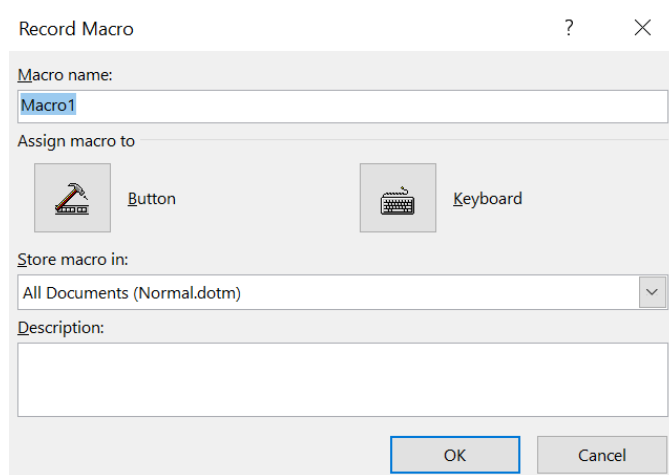
!!! Для знайомства з прикладами кодів доцільно скористатися прийомом, коли ви форматуєте частину документа в режимі запису макросу, а потім аналізуєте код макросу та редагуєте його.

Для запису макросу необхідно:

1. Знайти меню **Макрос**. Цей інструмент є як на вкладці **View**, так і на **Developer**



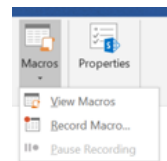
2. Ввести ім'я нового макросу



3. Заповнити поля (за необхідністю)

4. Ввести у полі **Assign macro to** або **Button** (Кнопку) (для призначення макросу кнопки панелі інструментів або команди меню або **Keyboard** (Клавішам) (для призначення макросу поєднання клавіш); Якщо макросу призначення не потрібне, натисніть кнопку ОК. Якщо в процесі роботи ви часто використовуєте якийсь макрос, то доцільно призначити кнопку або поєднання клавіш

6. Виконати дії, які потрібно включити до макросу



!!! Слід пам'ятати, що для запису дій треба використовувати тільки клавіатуру, оскільки дії, зроблені за допомогою миші, НЕ записуються

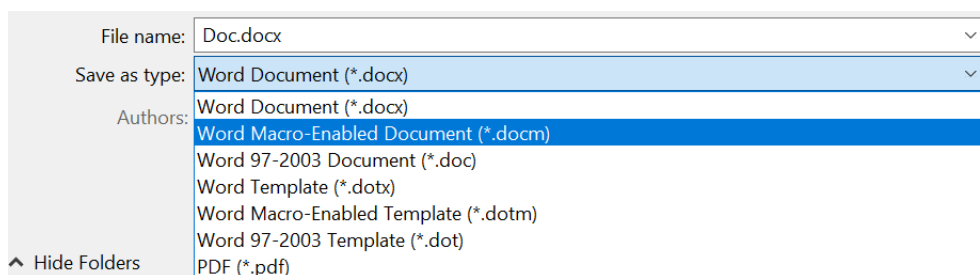
7. Після запису макросу треба

зупинити запис

Існує одна особливість, - при збереженні готового документа Word, в якому знаходяться макроси та процедури. Якщо ви згадаєте роботу з макросами в офісних додатках, то при відкритті готового документа там завжди ставить питання: "Відключати макроси при відкритті чи ні?". І якщо ви скажете, що макроси треба відключити, то підготовлені кнопки не працюватимуть 😞

У редакторі Word, на відміну від Excel, встановлено сильний захист від макросів (макроси розглядаються як віруси), і при повторному відкритті одного і того ж документа, в якому всі програми та макроси працювали, запиту на включення чи відключення макросів не буде, і працювати також нічого не буде. Тому при збереженні документа не забувайте про вкл/викл рівня безпеки

8. Збережіть документ. Документи з макросами зберігаються у спец.форматі ***.docm**



Завдання № 1 «Простий макрос»

Створити кнопки на документі, за допомогою яких можна провести аналіз речення. При встановленні курсору на знайдене слово та натисканні на відповідну клавішу слово має фарбуватись у певний колір та підкреслюватися.

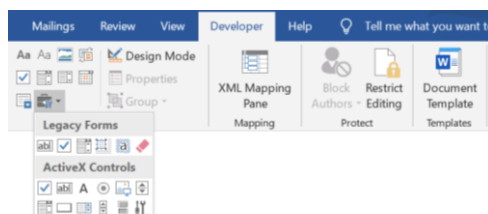
Порядок роботи.

1. Набрати/скопіювати довільний текст у редакторі Word
2. Обрати опцію «Макроси / Запис Макросу»
3. Надати макросу ім'я
4. Визначити в полі «Призначити макрос» спосіб, за допомогою якого викликатиметься макрос (кнопка або клавіші на вибір)
6. Виконати дії, які потрібно включити до макросу

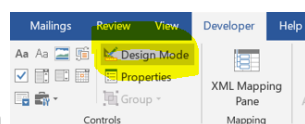
Послідовність дій така: встановити курсор перед будь-яким словом; виділити це слово; встановити колір шрифту – червоний, одинарне підкреслення.

Не забувайте, що для запису дій треба використовувати клавіатуру!!!

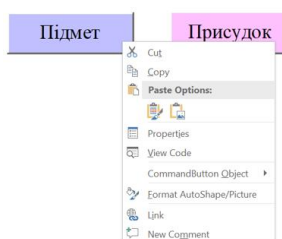
7. Натиснути після запису макросу кнопку «Зупинити запис»
8. Виконати ті ж дії для визначення присудка (*тільки колір шрифту встановити зелений і встановити подвійне підкреслення*).
9. Підготувати кнопку `CommandButton1` на робочому аркуші вище за текст, назвати її «Підмет»



10. У режимі Конструктора



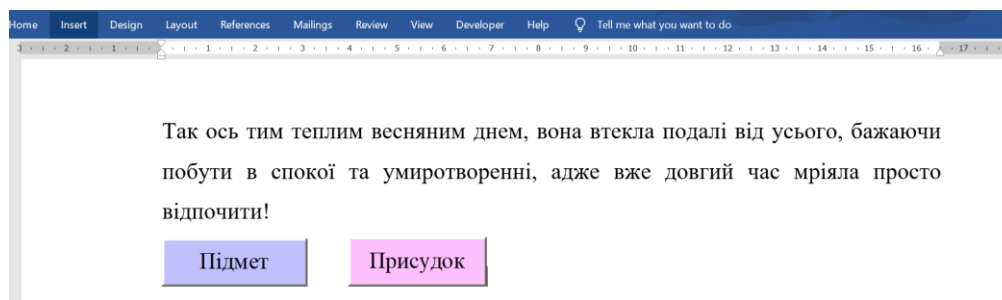
надати різні властивості цьому елементу



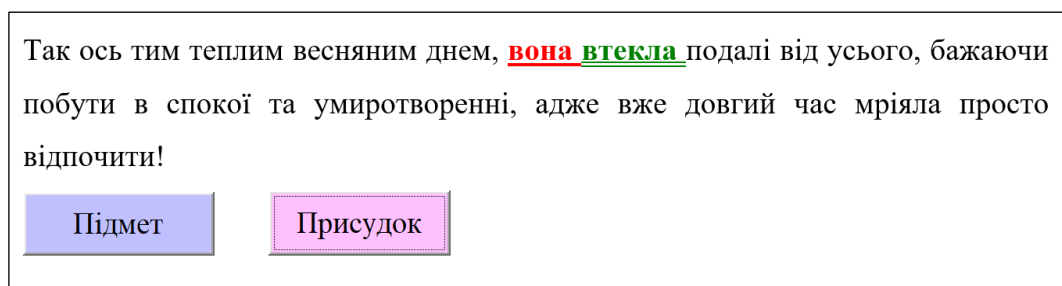
11. Переглянути записану програму, скопіювати її та переписати через режим написання програми Вихідний текст у кнопку «Присудок»
11. Виконати ті ж дії і для кнопки «Присудок»

12. Встановити рівень безпеки документа

13. Зберегти роботу. Вона виглядатиме приблизно так:



Після роботи кнопок:



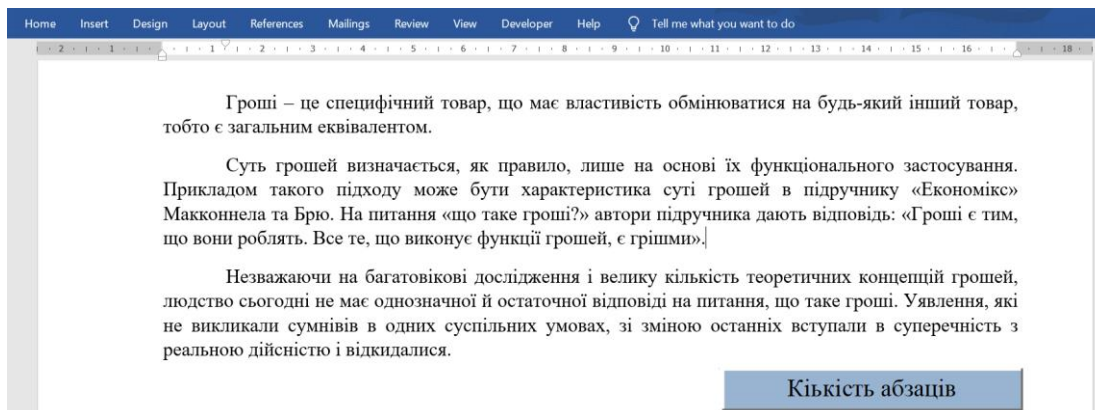
Завдання № 2 «Кількість абзаців в документі»

Створити/скопювати текст, що складається з кількох абзаців (від 5 до 10). Завдання у тому, щоб програмно визначити кількість абзаців у документі; озаглавити текст; наприкінці документа вивести повідомлення про кількість абзаців.

Ми знаємо, що в Word кількість абзаців визначається за символом ¶, який з'являється кожен раз після натискання на клавіші *Enter*. Але перевіряти та підраховувати кількість цих символів недоцільно, оскільки існують стандартні методи. Згадаймо, що документ, з яким ми працюємо, називається активним документом, і в програмі будемо звертатися до нього *ActiveDocument*.

Візьмемо для прикладу текст про Гроші.

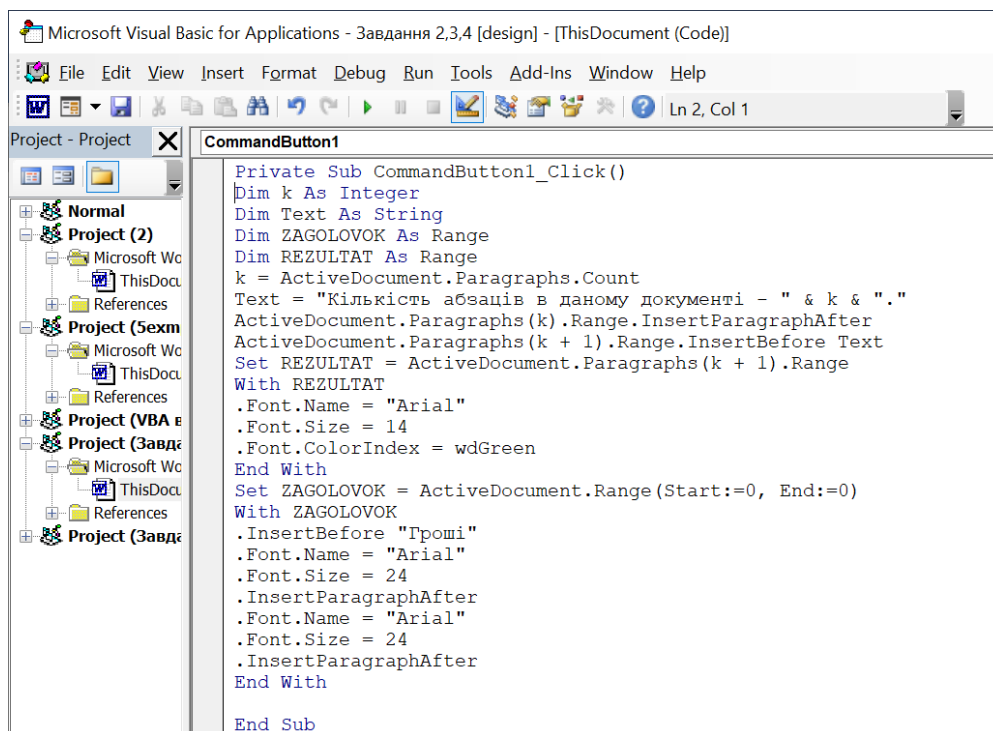
Текст містить 3 абзаци, судячи з кількості спеціальних символів.



Порядок роботи

Створити текст в документі. Поставити на документ кнопку «Кількість абзаців», перемістити її під текст та змінити дизайн

Після цього треба в контекстному меню обрати команду «Переглянути код», після чого відкриється редактор режиму написання коду програми



```
Private Sub CommandButton1_Click()  
Dim k As Integer  
Dim Text As String  
Dim ZAGOLOVOK As Range  
Dim REZULTAT As Range  
k = ActiveDocument.Paragraphs.Count  
Text = "Кількість абзаців в даному документі - " & k & "."  
ActiveDocument.Paragraphs(k).Range.InsertParagraphAfter  
ActiveDocument.Paragraphs(k + 1).Range.InsertBefore Text  
Set REZULTAT = ActiveDocument.Paragraphs(k + 1).Range  
With REZULTAT  
.Font.Name = "Arial"  
.Font.Size = 14  
.Font.ColorIndex = wdGreen  
End With  
Set ZAGOLOVOK = ActiveDocument.Range(Start:=0, End:=0)  
With ZAGOLOVOK  
.InsertBefore "Гроші"  
.Font.Name = "Arial"  
.Font.Size = 24  
.InsertParagraphAfter  
.Font.Name = "Arial"  
.Font.Size = 24  
.InsertParagraphAfter  
End With  
End Sub
```

Алгоритм виконання завдання:

1. Оголосити змінні:

- a) k - змінна для підрахунку кількості абзаців у документі;
- b) Text - змінна для формування тексту-відповіді про кількість абзаців;
- c) ZAGOLOVOK — змінна, яка визначає область для виведення заголовку

2. Визначити кількість абзаців за допомогою методу Count об'єкта Paragraphs

3. Сформувати текст виведення повідомлення

4. Вставити додатковий абзац після останнього абзацу для виведення повідомлення

5. Вивести текст-повідомлення до підготовленого абзацу

6. Сформувати область для виведення заголовка **Гроші**

7. Вивести назву **Гроші**

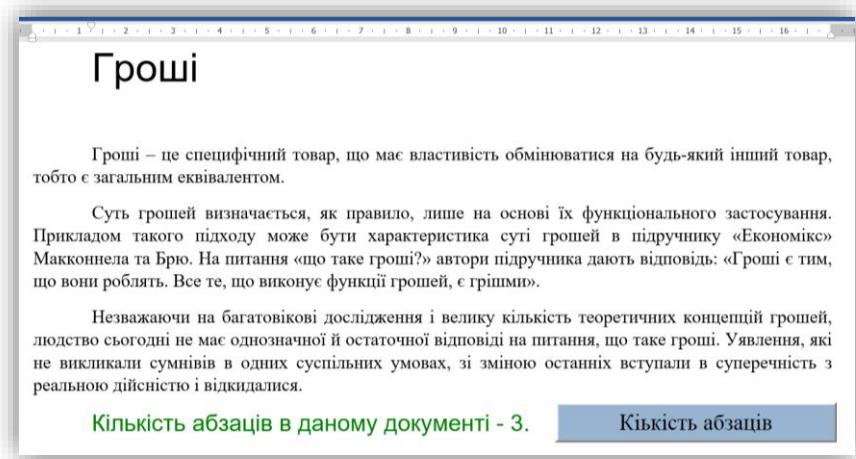
8. Встановити шрифт та розмір заголовка

9. Вставити символ кінця абзацу для того, щоб назва та початок тексту були на різних рядках.

Можна доповнити це завдання.

Наприклад, повідомлення про кількість абзаців нехай виводиться іншим кольором та розміром шрифту тощо

Результат виконання програми:



Текст програми

```
Public Sub CommandButton1_Click()  
    Dim k As Integer  
    Dim Text As String  
    Dim ZAGOLOVOK As Range  
    Dim REZULTAT As Range  
    k = ActiveDocument.Paragraphs.Count  
    Text = "Кількість абзаців в цьому документі - " & k & ". "  
    ActiveDocument.Paragraphs(k).Range.InsertParagraphAfter  
    ActiveDocument.Paragraphs(k + 1).Range.InsertBefore Text  
    Set REZULTAT = ActiveDocument.Paragraphs(k + 1).Range  
    With REZULTAT .Font.Name = "Arial" .Font.Size = 14  
        .Font.ColorIndex = wdDarkRed End With  
    Set ZAGOLOVOK = ActiveDocument.Range(Start:=0, End:=0)  
    With ZAGOLOVOK  
        .InsertBefore "Гроші"  
        .Font.Name = "Arial"  
        .Font.Size = 24  
        .InsertParagraphAfter  
        .Font.Name = "Arial"  
        .Font.Size = 24  
        .InsertParagraphAfter  
    End With  
End Sub
```

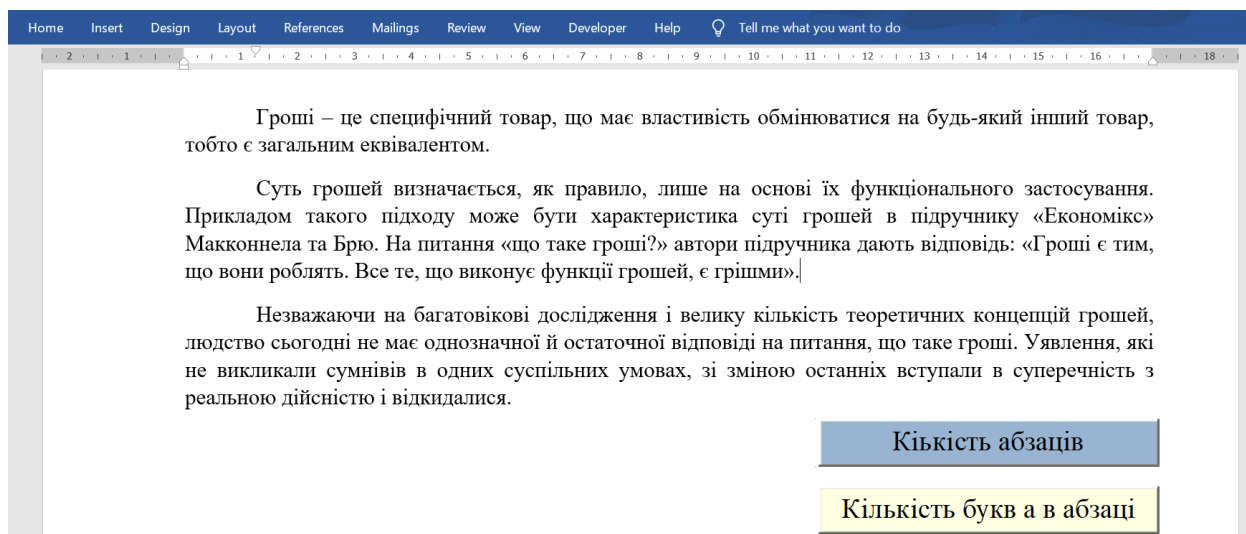
Завдання № 3 «Кількість букв «а» (укр) в абзаці»

Ця практична робота виконується з тим самим документом, що і попереднє завдання (про кількість абзаців).

Завдання у тому, щоб програмно визначити номер абзацу, у якому відбуватиметься підрахунок букв «а». Треба підрахувати їхню кількість, а наприкінці документа (або після абзацу, у якому відбувався підрахунок) вивести повідомлення про кількість літер.

Порядок роботи (аналогічний до попереднього завдання):

Створити на документі відповідну кнопку «Кількість букв «а» в абзаці», перемістити її під текст та змінити дизайн



Алгоритм виконання завдання наступний:

1. Оголосити змінні:

- a) b — рядкова змінна для розміщення у ній вибраного абзацу у документі;
- b) k – змінна для зберігання кількості абзаців у документі;
- c) dl – змінна для зберігання довжини абзацу;
- d) Text - змінна для формування тексту-відповіді про кількість символів а в абзаці;
- e) Nab - змінна для введення номера абзацу, в якому буде відбуватися підрахунок;
- f) I - змінна для організації циклу;
- g) kol - змінна для підрахунку символів а;
- h) REZULTAT - змінна, яка визначить область для виведення заголовка.

2. Введіть номер абзацу за допомогою діалогового вікна

3. Визначити кількість абзаців у документі

4. Перевірити наявність введеного номера абзацу та видати повідомлення про помилку, якщо такого немає

5. У змінну b занести текст абзацу

6. Визначити довжину абзацу

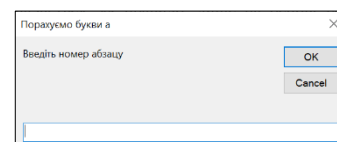
7. Організувати цикл знаходження символу а

8. Сформувати текст виведення повідомлення

9. Вставити додатковий абзац після останнього абзацу для виведення повідомлення

10. Вивести текст-повідомлення до підготовленого абзацу

11. Встановити певний шрифт, розмір та колір тексту-повідомлення



Результат виконання програми:

Гроші – це специфічний товар, що має властивість обмінюватися на будь-який інший товар, тобто є загальним еквівалентом.

Суть грошей визначається, як правило, лише на основі їх функціонального застосування. Прикладом такого підходу може бути характеристика суті грошей в підручнику «Економікс» Макконнела та Брю. На питання «що таке гроші?» автори підручника дають відповідь: «Гроші є тим, що вони роблять. Все те, що виконує функції грошей, є грошми».

Незважаючи на багатовікові дослідження і велику кількість теоретичних концепцій грошей, людство сьогодні не має однозначної й остаточної відповіді на питання, що таке гроші. Уявлення, які не викликали сумнівів в одних суспільних умовах, зі зміною останніх вступали в суперечність з реальною дійсністю і відкидалися.

Кількість абзаців

Кількість букв а в абзаці з номером 2 - 21.

Кількість букв а в абзаці

Текст програми

```
Private Sub CommandButton2_Click()  
    Dim b As String  
    Dim k As Integer  
    Dim dl As Long  
    Dim Text As String  
    Dim Nab As Integer  
    Dim i As Long  
    Dim REZULTAT As Range  
    kol = 0  
    Nab = InputBox("Введіть номер абзацу", "Порахуємо букви а")  
    k = ActiveDocument.Paragraphs.Count  
    If Nab > k Then  
        MsgBox "В тексті немає такого абзацу", 48, "Попередження"  
    End If  
    b = ActiveDocument.Paragraphs(Nab).Range  
    dl = Len(b)  
    For i = 1 To dl  
        If Mid(b, i, 1) = "a" Or Mid(b, i, 1) = "A" Then kol = kol + 1  
    Next i  
    MsgBox kol  
    Text = "Кількість букв а в абзаці з номером "& Nab & " - " & kol & "."  
    ActiveDocument.Paragraphs(k).Range.InsertParagraphAfter  
    Set REZULTAT = ActiveDocument.Paragraphs(k + 1).Range  
    With REZULTAT  
        .InsertBefore Text  
        .Font.Name = "Arial"  
        .Font.Size = 14  
        .Font.ColorIndex = wdDarkRed  
    End With  
End Sub
```

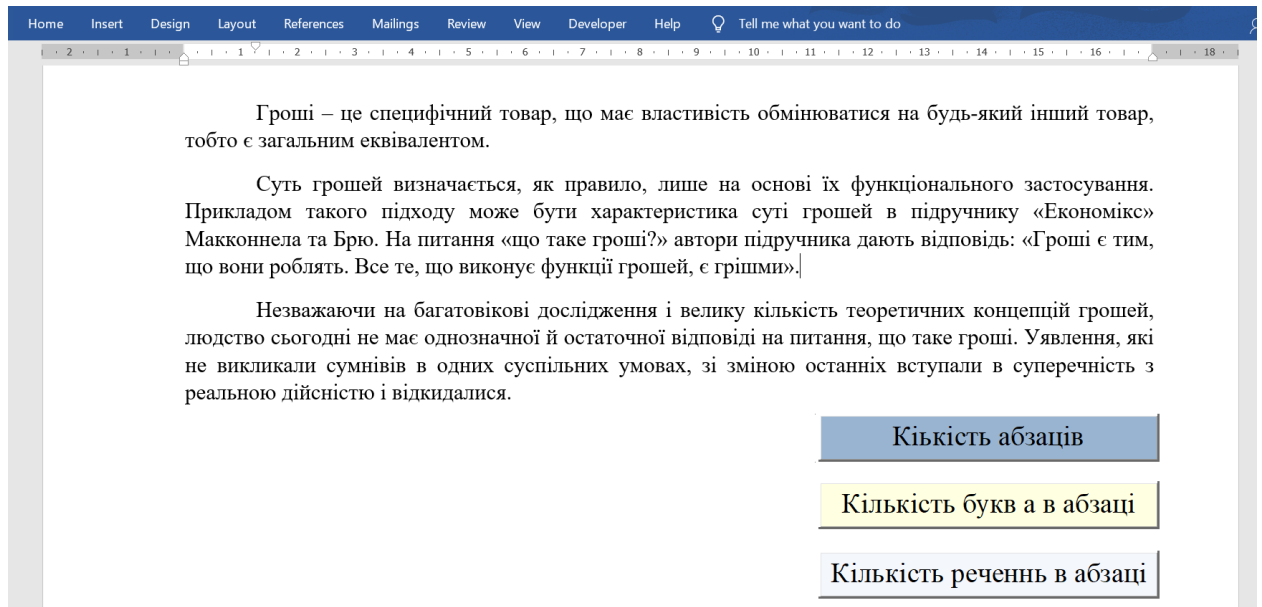
Завдання № 4 «Кількість речень в абзаці»

Ця практична робота виконується з тим самим документом, що й роботи 2 та 3 (або з будь-яким текстом, що складається з кількох абзаців).

Завдання полягає в тому, щоб програмно визначити номер абзацу, в якому є найбільша кількість речень; видати повідомлення про номер абзацу та про кількість речень, використовуючи діалогове вікно; виділити цей абзац іншим кольором.

Порядок роботи (аналогічний до попереднього завдання):

Створити на документі відповідну кнопку «Кількість речень в абзаці», перемістити її під текст та змінити дизайн



Алгоритм виконання завдання наступний:

1. Оголосити змінні:

- a) k – змінна для зберігання кількості абзаців у документі;
- b) kol - змінна для зберігання кількості пропозицій в абзаці;
- c) i - змінна для організації циклу;
- d) Mas() — масив для занесення кількості речень в абзацах;
- e) Otvet – змінна для формування тексту-відповіді з результатами;
- f) Max - змінна для зберігання максимальної кількості пропозицій в абзаці;
- g) ind – змінна для зберігання номера абзацу з максимальною кількістю речень;

h) REZULTAT – змінна, яка визначить область абзацу, до якого потрібно застосувати елементи форматування

При оголошенні масиву нам невідома кількість абзаців у документі. Тому оголошення масиву робимо в два етапи: 1-й етап - оголошення без розміру Dim Mas () As Integer; 2-й етап – після визначення кількості абзаців у документі перевизначаємо розмірність масиву ReDim Mas(k) As Integer. Такі масиви називають динамічними.

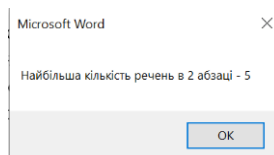
Крім того, щоб індекс масиву починався з 1, а не з 0 (для зручності роботи), у редакторі на початку аркуша з процедурою вкажіть команду Option Base 1.

2. Визначити кількість абзаців у документі
3. Перевизначити розмірність масиву
4. Організувати цикл визначення кількості речень в абзацах із занесенням їх розмірності в масив і збереженням порядкового номера в масиві, що відповідає номеру абзацу
5. Організувати цикл із знаходження найбільшої кількості речень в абзаці (робота зі сформованим масивом) за стандартним алгоритмом
6. Сформуванати текст виведення повідомлення
7. Вивести у діалоговому вікні повідомлень результат
8. Визначити область абзацу для форматування
9. Встановити певний шрифт, розмір та колір тексту-абзацу з максимальною кількістю речень.

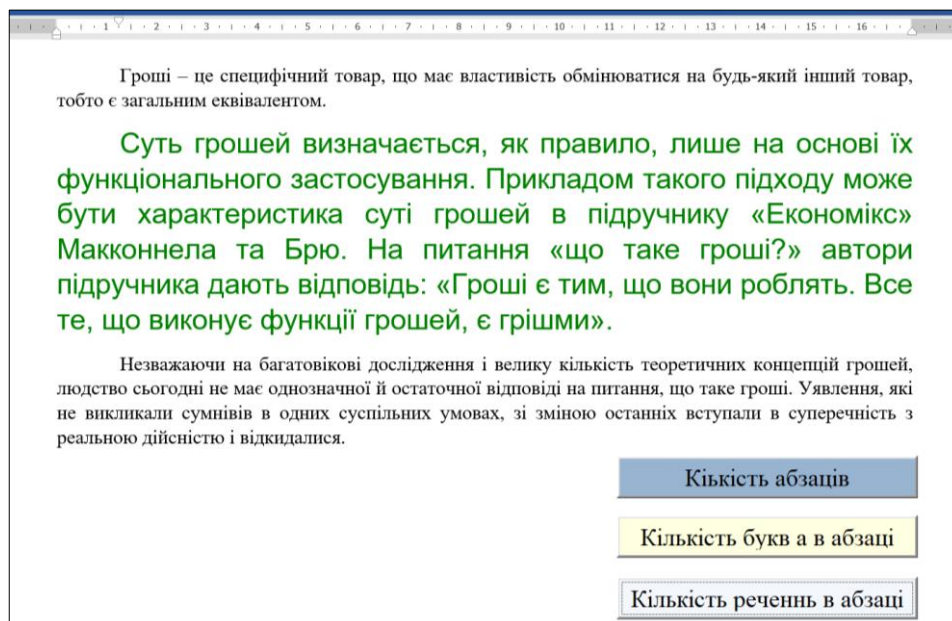
Текст програми

```
Public Sub CommandButton3_Click()
    Dim k As Integer
    Dim kol As Integer
    Dim i As Integer
    Dim Mas() As Integer
    Dim otvet As String
    Dim Max As Integer
    Dim ind As Byte
    Dim REZULTAT As Range
    kol = 0: k = 0
    k = ActiveDocument.Paragraphs.Count
    ReDim Mas(k) As Integer
    For i = 1 To k
        kol = ActiveDocument.Paragraphs(i).Range.Sentences.Count
        Mas(i) = kol
    Next i
    Max = Mas(1)
    ind = 1
    For i = 2 To k
        If Mas(i) > Max Then
            Max = Mas(i)
            ind = i
        End If
    Next i
    otvet = "Найбільша кількість речень в "& ind & " абзаці - " & Max
    MsgBox otvet
    Set REZULTAT = ActiveDocument.Paragraphs(ind).Range
    With REZULTAT
        .Font.Name = "Arial"
        .Font.Size = 24
        .Font.ColorIndex = wdDarkRed
    End With
End Sub
```

Діалогове вікно з результатом- повідомленням виглядатиме так:



Результат виконання програми:

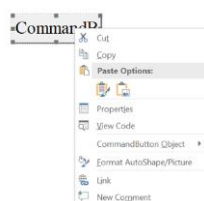


Завдання № 5 «Слова-паліндроми»

Завдання: програмно (код, а не макрорекордер) визначити, чи є в тексті слова-паліндроми та позначити ці слова іншим кольором.

Хід роботи

1. Підготувати текст зі словами- паліндромами ШАЛАШ, ДІД, ОКО тощо
2. Створити кнопку “CommandButton1” на робочому листі нижче тексту
3. Перейменувати кнопку в «Паліндроми»; змінити властивості оформлення (за власним бажанням); надати формат «обтекания по контуру», щоб її можна було переміщати по листу
4. Обрати в контекстному меню цього об’єкту пункт Code (переглянути код або вихідний код) або два рази клікнути по елементу



Алгоритм виконання завдання:

1. Оголосити змінні:

- a) Slovo — змінна для послідовного збереження виділених слів;
- b) L — змінна для збереження довжини слова;
- c) M — змінна, що визначає половину довжини слова, що виділено;
- d) K — змінна для формування 1-ї половини слова;
- e) D — змінна для формування 2-ї половини слова;
- f) Kol — змінна для збереження кількості абзаців в документі

2. Визначити кількість абзаців в документі

3. Визначити область, в котрій треба шукати слова-паліндроми (це всі абзаци тексту)

4. Організувати цикл по виділенню слів в області визначення

5. Привласнити виділене слово рядковій змінній

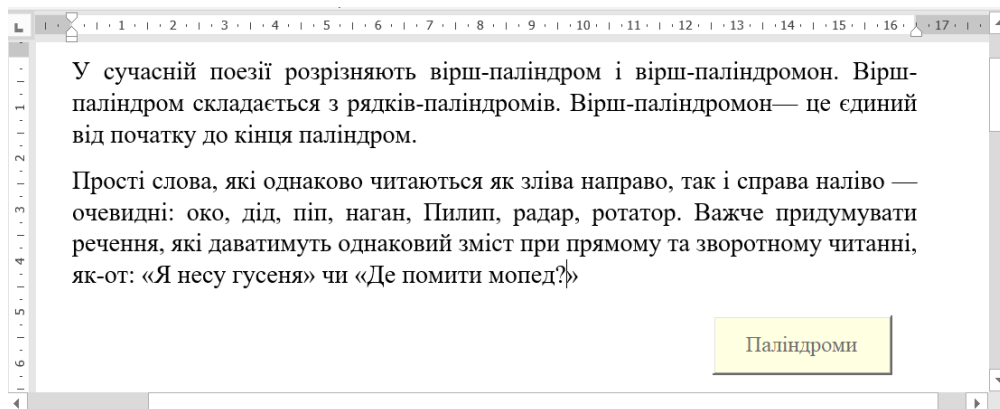
6. Визначити довжину виділеного слова

7. Перевірити довжину слова: якщо вона = 1, то слово перевіряти немає змісту

8. Організувати цикл, щоб визначити, чи є слово паліндромом (порівнюючи букви з різних половин слова)

9. Змінити колір букв у слові-паліндромі

Приклад



Прописуємо текст, розміщуємо елемент кнопка на документі; змінюємо його властивості: розмір, колір тощо. В режимі Конструктор прописуємо код для виконання `CommandButton1_Click()`

```
Private Sub CommandButton1_Click()
```

```
Dim slovo As String
```

```
Dim L As Byte
```

```
Dim M As Byte
```

```
Dim K As String
```

```
Dim D As String
```

```
Dim kol As Byte
```

```
kol = ActiveDocument.Paragraphs.Count
```

```
Set myRange = ActiveDocument.Range
```

```
For Each aword In myRange.Words
```

```
slovo = aword.Text
L = Len(slovo)
If L <= 1 Then GoTo M1
M = Int(L / 2)

For i = 1 To M
    K = Mid$(slovo, i, 1)
    D = Mid$(slovo, L - i + 1, 1)
    If K <> D Then GoTo M1
Next i

aword.Font.Size = 14
aword.Font.ColorIndex = wdDarkBlue
M1: Next aword
End Sub
```

* В результаті виконання програми слова-паліндроми будуть виділено синім кольором

Робота макросу:

У сучасній поезії розрізняють вірш-паліндром і вірш-паліндромон. Вірш-паліндром складається з рядків-паліндромів. Вірш-паліндромон— це єдиний від початку до кінця паліндром.

Прості слова, які однаково читаються як зліва направо, так і справа наліво — очевидні: **око**, **дід**, **піп**, **наган**, **Пилип**, **радар**, **ротатор**. Важче придумувати речення, які даватимуть однаковий зміст при прямому та зворотному читанні, як-от: «Я несу гусеня» чи «Де помити мопед?»

Паліндроми

Завдання (just for fun)

Приклад малювання графіки в MS Word VBA 😊

```
Sub hexagon()

Dim hexagon_width As Double
Dim hexagon_height As Double
Dim hexagon_xspacing As Double
Dim columns As Long
Dim rows As Long

columns = 10
rows = 10

hexagon_xspacing = ActiveDocument.PageSetup.PageWidth / columns
hexagon_width = (ActiveDocument.PageSetup.PageWidth / columns) / 3 * 2
hexagon_height = 2 * Sqr((hexagon_width / 2) ^ 2 - (hexagon_width / 4) ^ 2)

For counter_rows = 0 To rows - 1
    For counter_columns = 0 To columns - 1
        If counter_rows Mod 2 = 0 Then
            hexagon_offset = 0
        Else
            hexagon_offset = hexagon_width / 4 * 3
        End If

        With ActiveDocument.Shapes.AddShape(msoShapeHexagon, (counter_columns *
hexagon_xspacing) + hexagon_offset, counter_rows * (hexagon_height / 2), hexagon_width,
hexagon_height)

            .Adjustments(1) = 0.58001 / 2
            .Fill.ForeColor.ObjectThemeColor = Rnd() * 5 + 4
            .Line.ForeColor.RGB = RGB(255, 255, 255)
            .Line.Weight = 3
        End With
    Next counter_columns
Next counter_rows
```

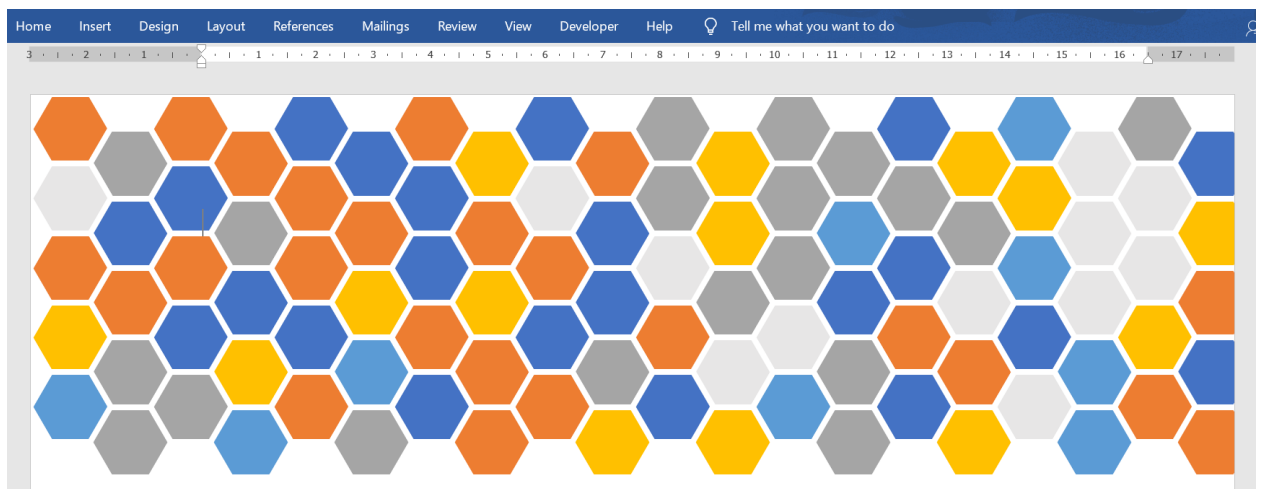
End With

Next counter_columns

Next counter_rows

End Sub

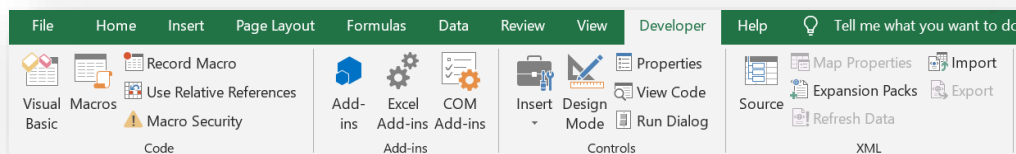
Результат:



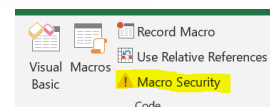
Комп'ютерний практикум №9

Створення макросів у додатку MS Excel (код на VBA)

На початку роботи в офісному додатку необхідно включити вкладку/ **Ribbon Developer tab** «Розробник» (Option – Customize Ribbon – Developer)

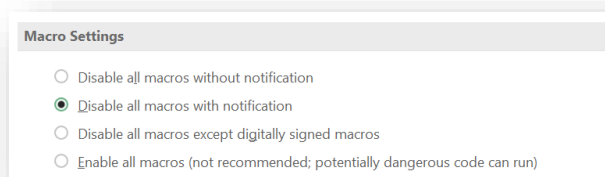


Після того, як вкладка Розробника буде активована, необхідно включити макроси. На вкладці Розробник знаходимо Безпеку макросів



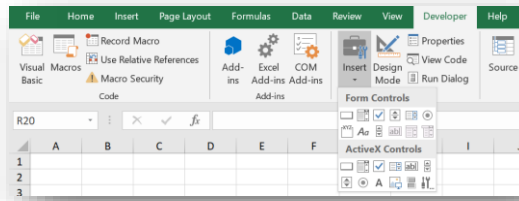
У вікні з налаштуваннями можна активувати відразу все макроси. Однак, слід звернути увагу на те, що розробники Microsoft не рекомендують вибирати цей варіант, так як є ймовірність запуску небезпечної програми, що може нашкодити комп'ютеру. Тому, виконуючи цю операцію, пам'ятайте, що ви дієте на свій страх і ризик. Рекомендується 2-й пункт ☺

5. **Вимкнути всі макроси без сповіщення:** вимикає всі макроси в документах та всі попередження про безпеку, пов'язані з макросами. Ця опція полягає у виборі, чи не довіряєте ніякому макросу.
6. **Вимкнути всі макроси за допомогою сповіщення:** це налаштування за замовчуванням. Це дозволяє активувати макроси у кожному конкретному випадку. Макроси вимкнені, але попередження про безпеку з'являються у присутності макросу.
7. **Вимкнути всі макроси, за виключенням цифрових підписаних макросів:** так само, як і в налаштуваннях за умовчанням, ця опція все ще дозволяє цифровим підписаним макросам, надісланим електронною поштою для запуску. Якщо це ще не так, ви отримаєте сповіщення, що дасть вам можливість це зробити;
8. **Увімкнути всі макроси:** цей параметр сильно не рекомендується, оскільки він показує комп'ютер, що виконує шкідливий код.



Далі можна розпочинати роботу з макросами.

Всі інструменти: кнопки, режим конструктора, в якому можна змінювати властивості, переглядати код тощо, знаходяться на вкладці Розробник



Завдання

Виконати **всі** приклади нижче, зібравши їх на одному листі Excel-файлу та оформивши за власним смаком (приклад оформлення рис.1), після чого виконати **Індивідуальне завдання** (на іншому листі цього ж файлу!!!) та написати до нього *Інструкцію* (приклад структури Інструкції Додаток А)

!!! Зберегти файл в форматі *.xlsm*

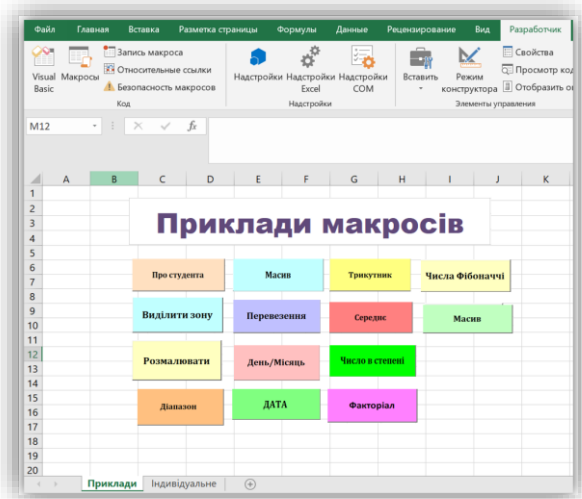
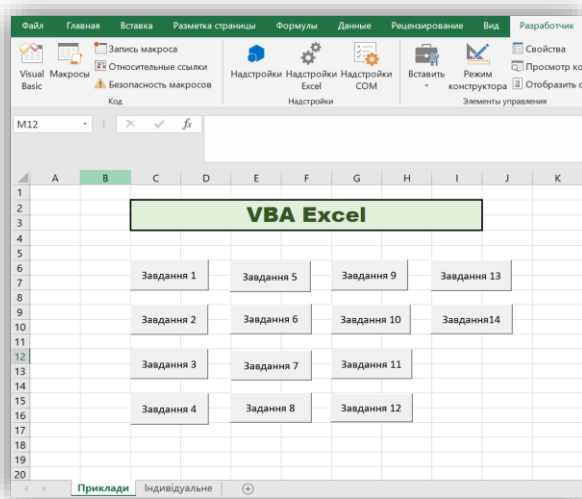


Рис 1. Приклади оформлення готового Excel-файлу завдання

Індивідуальне завдання до (VBA в Excel)

1. Придумати та реалізувати свій власний макрос в Excel, котрий використовує кожен з пунктів хоча б раз:
 - звернення до листів та комірок з даними
 - реалізацію певного обчислення
 - візуалізація результату (границі, колір, виділення тощо)
 - використання елементів форми
2. Створити власну змістовну користувацьку функцію в Excel (наприклад *тупате()*), яку можна буде викликати у рядку формул $= \text{тупате}()$ і отримувати результат обчислення

Задача №1

Додати на лист особисті дані: ім'я, прізвище, групу, факультет, рідне місто та вивести їх на екран

Рішення

Для цієї програми нам знадобиться стільки змінних, скільки забажаєте. Головне коректно ввести дані, а потім правильно вставити всі дані у відповідь (MsgBox). Т.ч. весь текст повинен бути в “ ”, а всі змінні повинні бути з'єднані символом &. Не забуваємо про пробіли.)

Власне, ось і вся програма 😊

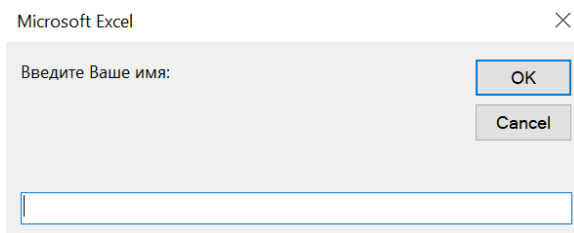
```
Private Sub CommandButton1_Click()  
Dim name, surname, group, dep, city As String
```

```
name = InputBox("Введіть Ваше ім'я: ")  
surname = InputBox("Введіть Ваше прізвище: ")  
group = InputBox("Введіть Вашу групу: ")  
dep = InputBox("Ви вчитесь на факультеті ... ")  
city = InputBox("Ваше рідне місто: ")
```

```
MsgBox ("Привіт, " & name & surname & " . Ти є почесним членом групи " &  
group & ",факультету " & dep & " . Роби щось для своєї групи! Місто " & city & "  
повинно пишатися тобою!!! ")
```

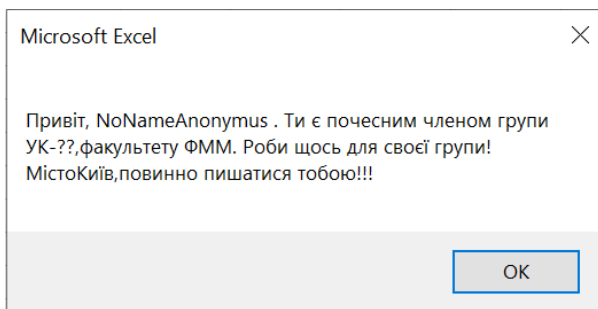
End Sub

Форми для введення даних будуть подібного типу:



...і так всі інші поля для вводу

Приклад результату:



Задача №2

Виділити зону

(Ввести початкову та кінцеву координати через VBA)

Рішення

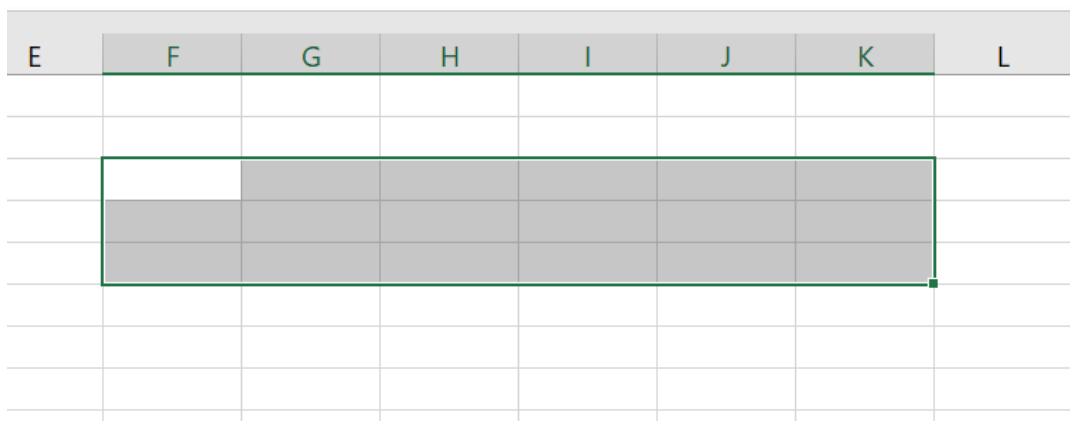
Почнемо з того, що створимо кнопку, яка буде реалізувати наше завдання, і визначимо основні дії.

Створюємо *InputBox*, через який вводимо потрібні нам дані. Потім через *Range(). Select* ми вказуємо, що задану зону потрібно виділити і завершуємо програму:

Код:

```
Private Sub CommandButton1_Click()  
Dim a As String  
Dim b As String  
Dim s As String  
a = InputBox(" Введіть початкову координату:")  
b = InputBox(" Введіть кінцеву координату:")  
s = a & ":" & b  
  
Range(s).Select  
  
End Sub
```

Наприклад, ввівши **F3** (початкова) та **K5** (кінцева), отримуємо



The image shows a portion of an Excel spreadsheet. The columns are labeled E, F, G, H, I, J, K, and L. The rows are numbered 1 through 5. A green border highlights a rectangular range of cells starting from cell F3 and extending to cell K5. This range includes cells F3, G3, H3, I3, J3, K3, F4, G4, H4, I4, J4, K4, F5, G5, H5, I5, J5, and K5.

E	F	G	H	I	J	K	L

Задача №5

Ввести 2-вимірний масив, в якому головна діагональ змінює значення на 0.

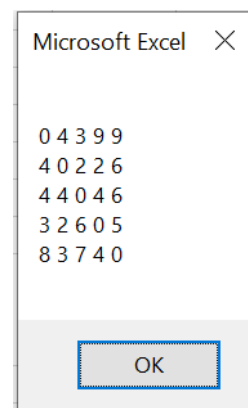
Рішення

Оголошуємо масив (`mass(10,10)`), далі **n** – кількість рядків та стовпців (адже головна діагональ лише у квадратних матрицях, **i** та **j** – лічильники по рядка і стовпцях, і **str** –зміну для виводу масиву в рядок. Заповнивши масив випадковими числами, визначаємо елементи головної діагоналі матриці. Це ті елементи, у яких співпадає номери рядку та стовпчика (тобто $i = j$) . Всі числа матриці, що залишилися, ми залишаємо незмінними. Виводимо результат. `vbCrLf` – переведення на новий рядок.

Код:

```
Private Sub CommandButton1_Click()  
Dim mass(100, 100) As Integer  
Dim m As Integer  
Dim j As Integer  
Dim n As Integer  
Dim i As Integer  
Dim str As String  
  
str = ""  
n = CInt(InputBox("Введіть розмірність матриці n:"))  
  
For i = 1 To n  
For j = 1 To n  
mass(i, j) = (Rnd * 8 + 1)  
  
If i = j Then mass(i, j) = 0  
str = str & mass(i, j) & " "  
Next j  
str = str & vbCrLf  
Next i  
MsgBox (str)  
  
End Sub
```

Наприклад, для розмірності 5 x 5:



Задача №6

Є певна кількість товару та машина, яка має його перевезти за один раз.

Визначити можливість такого перевезення

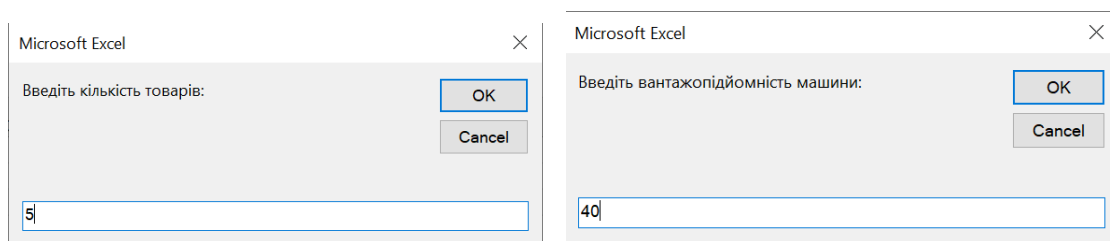
(Кількість, вага вантажу та вантажопідйомність машини задаємо самі!)

Рішення

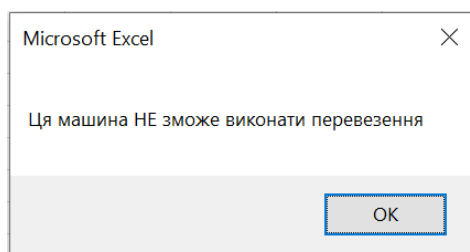
Для цього завдання нам знадобиться масив товарів (**mass(100)**), а також такі змінні як **n** (кількість товару в кг.), **gruz** (вантажопідйомність машини), **summ** (сума всього вантажу) та **i** (лічильник, який допоможе позначити номери вантажів). Щоб не було ускладнень **mass**, **gruz** та **summ** краще описати, як тип *Double*. Вводимо ваги товарів, підраховуємо загальну суму (вага всього вантажу), після чого порівнюємо її з вантажопідйомністю машини.

Код:

```
Private Sub CommandButton1_Click()  
    Dim mass(100) As Double  
    Dim gruz As Double  
    Dim n As Integer  
    Dim summ As Double  
    Dim i As Integer  
    n = CInt(InputBox("Введіть кількість товарів:"))  
    gruz = CInt(InputBox("Введіть вантажопідйомність машини:"))  
    summ = 0  
    For i = 1 To n  
        mass(i) = CDbI(InputBox(" Введіть вагу " & i & "-го товару"))  
        summ = summ + mass(i)  
    Next i  
  
    If summ < gruz Then MsgBox ("Ця машина зможе виконати перевезення") Else  
        MsgBox ("Ця машина НЕ зможе виконати перевезення")  
    End Sub
```



Далі вводимо всі товари вагою по 10 (загальна сума 50 > 40)



Задача №7

Дано 2 цілих числа: *D* (день) та *M* (місяць), що визначають коректну дату невисокосного року. Вивести значення *D* і *M* для дати, що передує вказаній

Рішення

По-перше, нам необхідно ввести чотири змінні. Далі нам треба ввести дату (день/місяць), попередню від якої ми хочемо дізнатися (D/M). (Функція *Cdbl* перетворює вираз у тип *Double*)

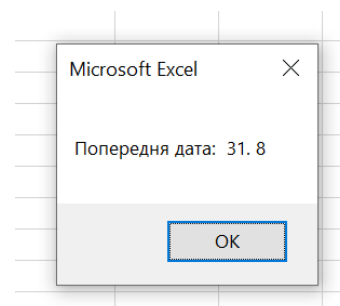
Задаємо умову для перевірки: що означає, що якщо число попереднього дня дорівнюватиме 0, - нам просто потрібно поставити попередній місяць, а от якщо число шуканого місяця дорівнюватиме 0, то замість 0 ставимо 12.

Далі треба розглянути Випадок (Case) для кожного місяця окремо (бо кількість днів у місяцях різна), використовуємо оператор *Select*.

Код:

```
Private Sub CommandButton1_Click()  
Dim d, m, D_pred, M_pred As Integer  
d = Cdbl(InputBox("Введіть D "))  
m = Cdbl(InputBox("Введіть M "))  
D_pred = d - 1  
If D_pred = 0 Then  
M_pred = m - 1  
If M_pred = 0 Then M_pred = 12  
Select Case M_pred  
Case 1  
D_pred = 31  
Case 2  
D_pred = 28  
Case 3  
D_pred = 31  
Case 4  
D_pred = 30  
Case 5  
D_pred = 31  
Case 6  
D_pred = 30  
Case 7  
D_pred = 31  
Case 8  
D_pred = 31  
Case 9  
D_pred = 30  
Case 10  
D_pred = 31  
Case 11  
D_pred = 30  
Case 12  
D_pred = 31  
End Select  
Else  
M_pred = m  
End If  
MsgBox ("Попередня дата ' " & D_pred & "." & M_pred)  
  
End Sub
```

Наприклад, для дати 1 вересня (вводимо як 1 та 9):



Задача №8

Ввести дату цифрами, а вивести її на екран словами

Рішення

По суті, це завдання дуже просте, але при написанні потрібно багато механічної роботи.

Знадобиться нам лише 4 змінні: **m** (ввести номер місяця), **d** (ввести номер дня), **Month** (вивести назву місяця словами) та **Day** (вивести номер дня словами). Month та Day будуть виведені через функцію String:

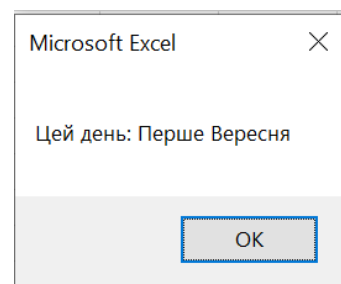
Код:

```
Private Sub CommandButton1_Click()  
Dim m As Integer  
Dim Month As String  
Dim d As Integer  
Dim Day As String  
m = CInt(InputBox("Введіть номер місяця:"))  
d = CInt(InputBox("Введіть номер дня:"))  
Select Case m  
Case 1  
Month = "Січня"  
Case 2  
Month = "Лютого"  
Case 3  
Month = "Березня"  
Case 4  
Month = "Квітня"  
Case 5  
Month = "Мая"  
Case 6  
Month = "Червня"  
Case 7  
Month = "Липня"  
Case 8  
Month = "Серпня"  
Case 9  
Month = "Вересня"  
Case 10  
Month = "Жовтня"  
Case 11  
Month = "Листопада"  
Case 12  
Month = "Грудня"  
End Select  
Select Case d  
Case 1  
Day = "Перше"  
Case 2  
Day = "Друге"  
Case 3  
Day = "Третє"  
Case 4  
Day = "Четверте"  
Case 5  
Day = "П'яте"  
Case 6  
Day = "Шосте"  
Case 7  
Day = "Сьоме"
```

```
Case 8
Day = "Восьме"
Case 9
Day = "Дев'яте"
Case 10
Day = "Десяте"
Case 11
Day = "Одинадцять"
Case 13
Day = "Дванадцять"
Case 13
Day = "Тринадцяте"
Case 14
Day = "Чотирнадцять"
Case 15
Day = "П'ятнадцять"
Case 16
Day = "Шістнадцять"
Case 17
Day = "Сімнадцять"
Case 18
Day = "Вісімнадцять"
Case 19
Day = "Дев'ятнадцять"
Case 20
Day = "Двадцять"
Case 21
Day = "Двадцять перше"
Case 22
Day = "Двадцять друге"
Case 23
Day = "Двадцять третє"
Case 24
Day = "Двадцять четверте"
Case 25
Day = "Двадцять п'яте"
Case 26
Day = "Двадцять шосте"
Case 27
Day = "Двадцять сьоме"
Case 28
Day = "Двадцять восьме"
Case 29
Day = "Двадцять дев'яте"
Case 30
Day = "Тридцять"
Case 31
Day = "Тридцять перше"
End Select
```

```
MsgBox ("Цей день: " & Day & " " & Month)
End Sub
```

Наприклад, для дати 1 вересня (вводимо як 1 та 9):



Задача №9

Елементи рівнобедреного прямокутного трикутника пронумеровані так:

1 – катет a ; 2 – гіпотенуза; 3 – висота h , опущена на гіпотенузу ($h=c/2$);

*4 – площа $S=c*h/2$, де $c = a \cdot \sqrt{2}$*

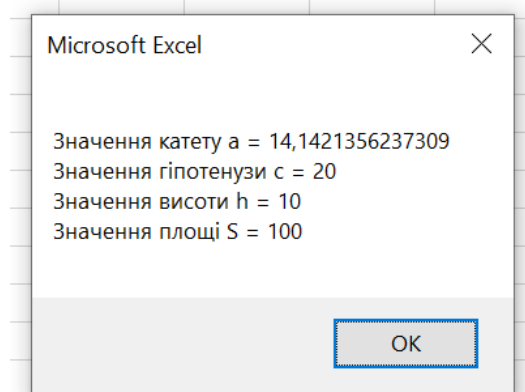
Дано номер одного з цих елементів та його значення. Вивести значення інших елементів цього трикутника (у тому самому порядку !!!)

Рішення

Код:

```
Private Sub CommandButton1_Click()  
Dim n As Integer  
Dim x, a, c, H, s As Double  
  
n = CDbI(InputBox("Введіть номер параметра N "))  
x = CDbI(InputBox("Введіть значення " & n & "-го параметру "))  
Select Case n  
Case 1  
a = x  
Case 2  
a = x / Sqr(2)  
Case 3  
a = 2 * x / Sqr(2)  
Case 4  
a = Sqr(2 * x)  
End Select  
  
c = a * Sqr(2)  
H = c / 2  
s = c * H / 2  
MsgBox ("Значення катету a = " & a & vbCrLf & "Значення гіпотенузи c = " & c &  
vbCrLf & "Значення висоти h = " & H & vbCrLf & "Значення площі S = " & s)  
  
End Sub
```

Наприклад, для значення вибору **2** (гіпотенуза) 20 результат буде:



Задача №10

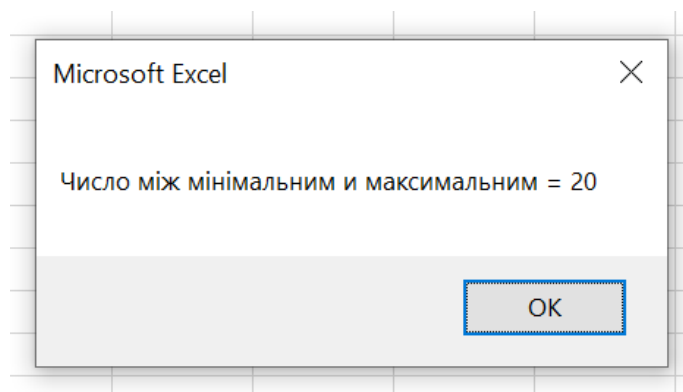
Дано 3 числа. Знайти число, що має середнє значення (розташовано між min і max)

Рішення

Код:

```
Private Sub CommandButton1_Click()  
Dim a As Integer  
Dim b As Integer  
Dim c As Integer  
Dim x As Integer  
  
a = CInt(InputBox("Введіть число A "))  
b = CInt(InputBox("Введіть число B "))  
c = CInt(InputBox("Введіть число C "))  
If (a < b) And (a > c) Or (a > b) And (a < c) Then x = a  
If (b < a) And (b > c) Or (b > a) And (b < c) Then x = b  
If (c < a) And (c > b) Or (c > a) And (c < b) Then x = c  
MsgBox ("Число між мінімальним и максимальним = " & x)  
End Sub
```

Наприклад, для набору значень 20, 50, -5 результат буде:



Задача №11

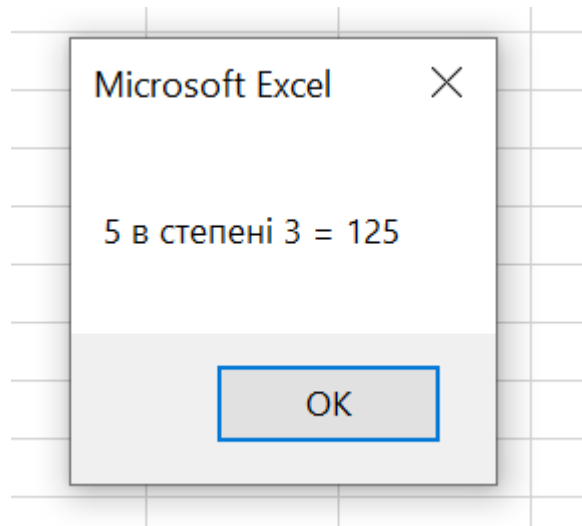
Дано дійсне число A и ціле число $N (>0)$. Знайти A^N .

Рішення

Код:

```
Private Sub CommandButton1_Click()  
    Dim f, a, n, i As Integer  
  
    a = CInt(InputBox("Введіть число A "))  
    n = CInt(InputBox("Введіть число N "))  
    f = 1  
    For i = 1 To n  
        f = f * a  
    Next i  
    MsgBox (a & " в степені " & n & " = " & f)  
End Sub
```

Наприклад, для значень 5 і 3:



Задача №12

Знайти факторіал заданого числа

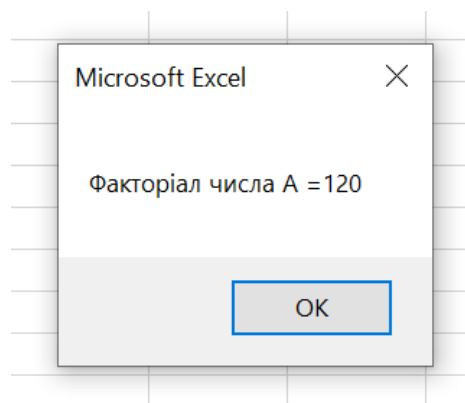
Рішення

Для вирішення цього завдання нам знадобиться три змінні: **a** (що відповідає за те число, факторіал якого ми хочемо знайти), **b** (допоміжна змінна, над якою будуть проводитися дії) і **f** (що відповідає за факторіал, - числа повинні перемножуватися). Також потрібно обов'язково задати змінним *b* і *f* їх початкові значення, щоб вони НЕ дорівнювали 0

Код:

```
Private Sub CommandButton1_Click()  
    Dim a As Integer  
    Dim b As Double  
    Dim f As Integer  
  
    f = 1  
    b = 1  
  
    a = CInt(InputBox("Введіть число A"))  
    For f = 1 To a  
        b = b * f  
    Next f  
    MsgBox (" Факторіал числа A =" & b)  
End Sub
```

Наприклад, для числа 5 факторіал:



Задача № 13

Дано ціле число $N (>1)$, яке є числом Фібоначчі:

$N = F_k$ (число Фібоначчі: $F_1=1, F_2=1, F_k=(F_{k-2}+F_{k-1}), k=3,4,\dots$).

Знайти цілі числа F_{k-1} та F_{k+1} – попереднє та наступне числа Фібоначчі.

(Наприклад: перші числа ряду Фібоначчі: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233,

Рішення

Для вирішення цього завдання знадобиться 4 змінних: F0 (перше число Фібоначчі), F1 (друге число Фібоначчі), f (формула числа Фібоначчі) та n (наше число, перед і після якого ми хочемо знайти число Фібоначчі). Оскільки всі наші 4 змінні цифри, поставимо Integer. І введемо наше власне число:

Код:

```
Private Sub CommandButton1_Click()
```

```
Dim F0, F1, f, n As Integer
```

```
n = CDbI(InputBox("Введіть число N "))
```

```
f = 0
```

```
F0 = 0
```

```
F1 = 1
```

```
Do While f < n
```

```
f = F1 + F0
```

```
F0 = F1
```

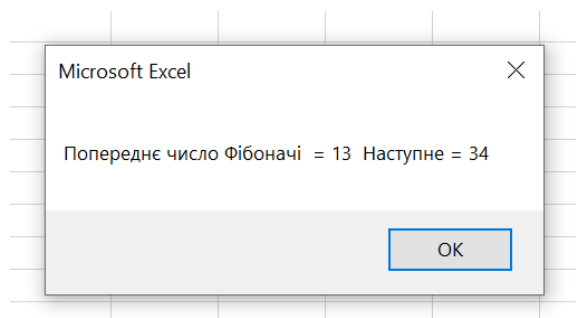
```
F1 = f
```

```
Loop
```

```
MsgBox ("Попереднє число Фібоначчі = " & F0 & " Наступне = " & F0 + F1)
```

```
End Sub
```

Наприклад, для числа 21 результат буде:



Задача №14

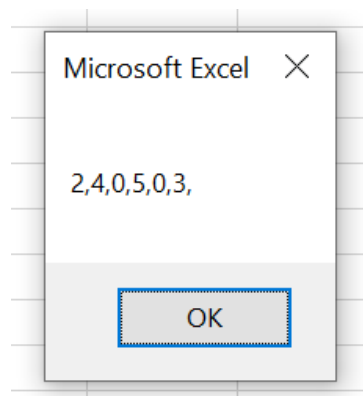
Дано цілочисельний масив з N елементів. Видалити з масиву всі однакові елементи (замінити їх нулями), залишивши їх перше входження

Рішення

Код:

```
Private Sub CommandButton1_Click()  
    Dim i, j, n, k As Integer  
    Dim a(50) As Integer  
    Dim str As String  
    n = CDbI(InputBox("Введіть розмірність масиву: "))  
    For i = 1 To n  
        a(i) = CDbI(InputBox("Введіть " & i & "-й елемент масиву"))  
    Next i  
    For i = 1 To n - 1  
        For j = i + 1 To n  
            If a(i) = a(j) Then a(j) = 0  
        Next j  
    Next i  
    str = ""  
    For i = 1 To n  
        str = str + CStr(a(i)) + ","  
    Next i  
    MsgBox (str)  
End Sub
```

Наприклад, для масиву 2 4 2 5 2 3 результат буде:



Структура Інструкції до індивідуального завдання
(.doc або .pdf файл)

Індивідуальне завдання
(1 частина)

Постановка задачі:

Описується суть завдання з усіма деталями

Хід роботи:

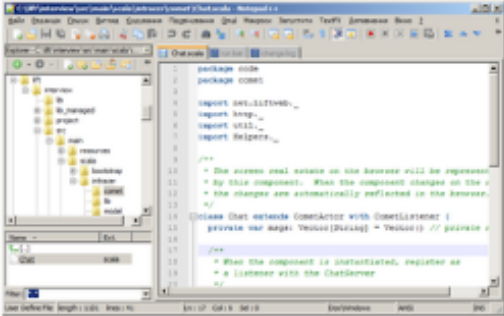

Описується загальне рішення, скріни роботи програми, код, проміжні результати тощо

Список використаних джерел

1. Стець О.В. Дистанційний курс: Інформатика 1. Прикладна інформатика в економіці URL: <https://classroom.google.com/c/NzIzMdc5OTU1Nzk1?cjc>
2. Стець О.В. Інформатика 1. Прикладна інформатика в економіці: Конспект лекцій [Електронний ресурс] : навч. посіб. для здобувачів ступеня бакалавр за спец. 051 Економіка / КПІ ім. Ігоря Сікорського ; уклад.: О.В.Стець. – Електрон. текст. дані (1 файл). – Київ : КПІ ім. Ігоря Сікорського, 2025.
3. Стець О.В. Методичні рекомендації до виконання комп.практ з дисципліни «Інформатика 1:«Розв'язування економічних задач засобами MS Excel» – Київ : КПІ ім. Ігоря Сікорського, 2019. 88 с. URL: <https://ela.kpi.ua/handle/123456789/47899>
4. Навчальні курси з Excel URL: <https://support.microsoft.com/uk-ua>
5. Бізнес-аналітика в Excel і служби Excel Services URL: <https://support.microsoft.com/uk-ua/office/>
6. LABA - —міжнародна бізнес-школа URL: <https://laba.ua/>
7. Віртуальний кабінет інформатики URL: <https://sites.google.com/view/vpu406/>

ДОДАТОК А

Налаштування NotePad++



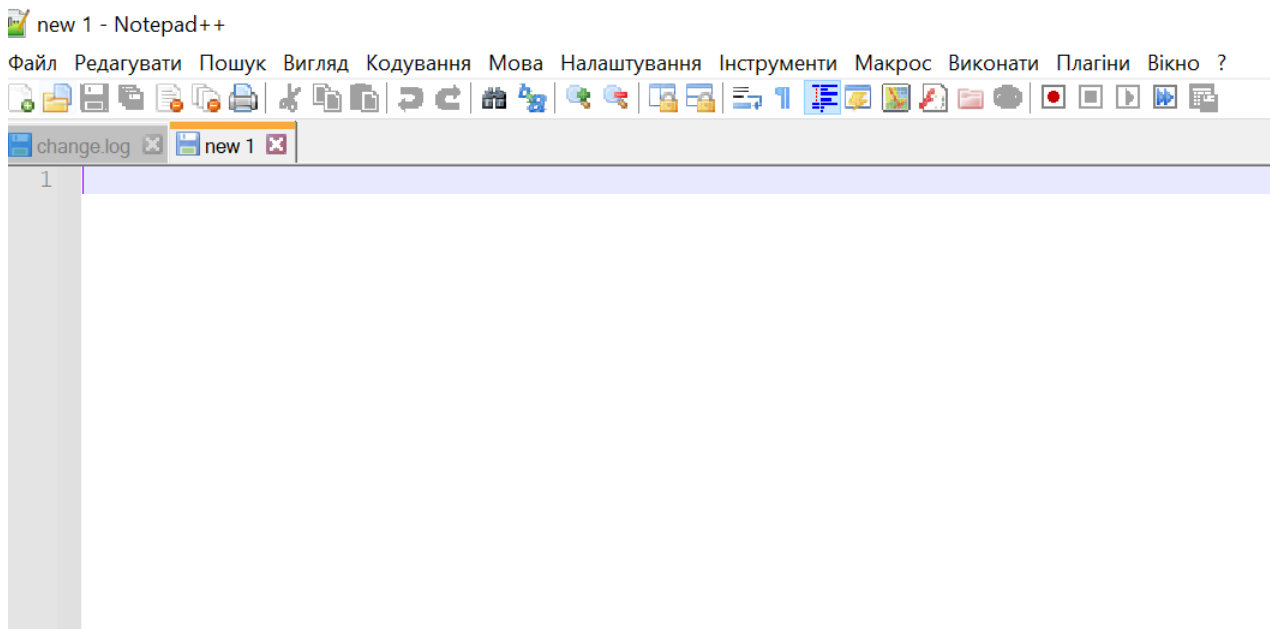
Тип	текстовий редактор
Автор	Don Ho
Розробник	Don Ho ^[1]
Перший випуск	24 листопада 2003
Стабільний випуск	8.6.1 (6 січня 2024; 28 днів тому ^[2])
Платформа	x86, x86_64 і Windows
Операційна система	Windows
Мова програмування	C++
Доступні мови	76 мов ^[3]
Українська мова	так
Стан розробки	активний
Ліцензія	GNU GPL Версія 2
Репозиторій	github.com/notepad-plus-plus/notepad-plus-plus ^[4]
Вебсайт	notepad-plus-plus.org ^[5]

Notepad++ — [текстовий редактор](#), призначений для програмістів і тих, кого не влаштовує скромна функціональність програми «блокнот», що входить до складу [Windows](#). **Notepad++** базується на компоненті Scintilla (потужному компоненті для редагування), написаному на [C++](#) з використанням тільки [Windows API](#) і [STL](#), що забезпечує максимальну швидкість роботи при мінімальному розмірі програми. Інтерфейс **Notepad++** є багатомовним (українська мова присутня). Серед особливостей програми — підсвічування синтаксису та підтримка великої кількості мов програмування ([C](#), [C++](#), [Java](#), [XML](#), [HTML](#), [PHP](#), [JavaScript](#), [ASCII](#), [Visual Basic / VBScript](#), [SQL](#), [Ruby](#), [CSS](#), [Pascal](#), [Perl](#) і [Python](#)), багатомовна підтримка, робота з декількома документами.

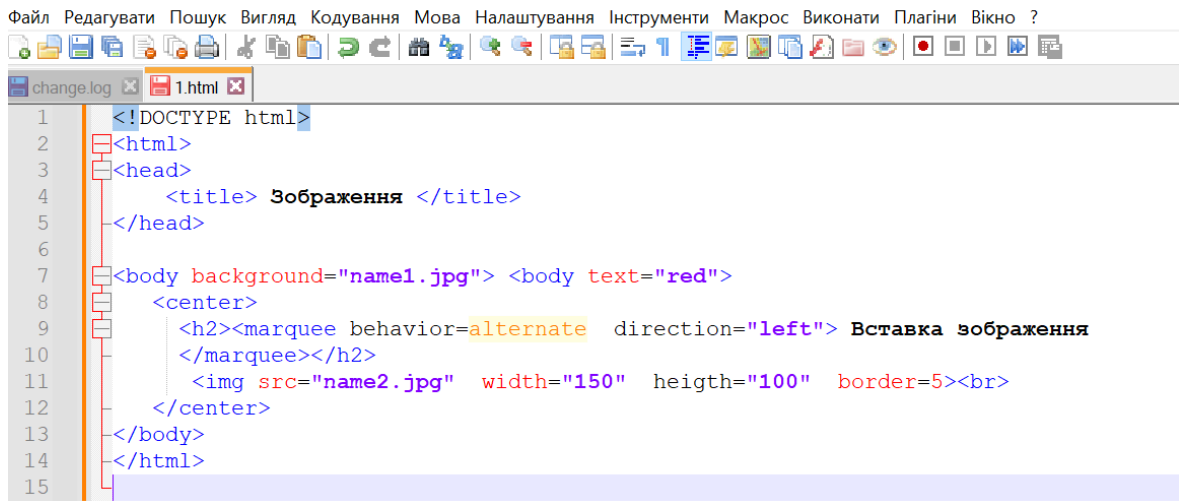
Основні можливості Notepad++ [\[ред. | ред. код \]](#)

- Підсвітка синтаксису тексту залежно від мови програмування, режим якої може налаштувати користувач;
- Можливість згортання блоків згідно з синтаксисом мови програмування;
- [WYSIWYG](#) (друкуєш і отримуєш те, що бачиш на екрані);
- Авто-завершення слова, що набирається;
- Одночасна робота з безліччю документів;
- Підтримка [регулярних виразів](#) для пошуку й заміни;
- Повна підтримка перетягування фрагментів тексту;
- Динамічна зміна вікон перегляду;
- Автоматичне визначення стану файлу;
- Збільшення і зменшення;
- Підтримка великої кількості мов;
- Нотатки;
- [Плагіни](#);
- Запис [макросу](#) і його виконання.

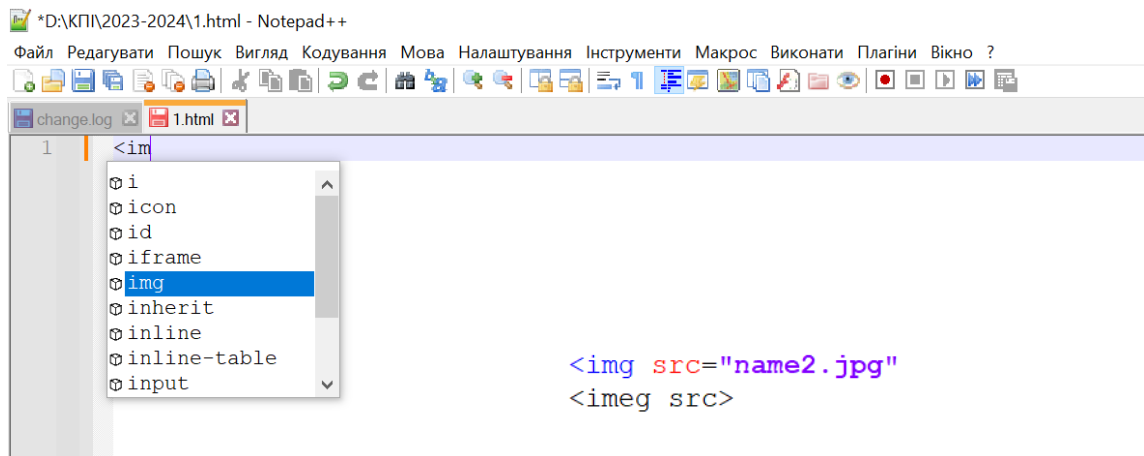
<https://notepad-plus-plus.org/>



NotePad++ підтримує форматування, групування, колірну схему, підказки тощо



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title> Зображення </title>
5 </head>
6
7 <body background="name1.jpg" <body text="red">
8   <center>
9     <h2><marquee behavior=alternate direction="left"> Вставка зображення
10    </marquee></h2>
11     <br>
12   </center>
13 </body>
14 </html>
15
```



```
*D:\КП\2023-2024\1.html - Notepad++
Файл Редагувати Пошук Вигляд Кодування Мова Налаштування Інструменти Макрос Виконати Плагіни Вікно ?
change.log 1.html
1 
```

Підключення браузерів

У більш ранніх версіях, браузери в редакторі були включені за замовчуванням, а тепер їх доводиться підключати вручну.

З чим це пов'язано? Можливо з безпекою. Адже був же злом версії 7.3.3, після чого і пропали браузери з меню «Запуск».

Тепер в меню «Запуск» немає списку браузерів.

Отже, спочатку напишемо що небудь в редакторі для тестування, після установки браузера.

*C:\Users\alenk\Desktop\new 1.html - Notepad++

Файл Редагувати Пошук Вигляд Кодування Мова Налаштування Інструменти Макрос Виконати Плагіни Вікно ?

```
change.log x new 1.html x
1 <html>
2 <head>
3   <title> NotePad++ </title>
4 </head>
5
6 <body>
7   Налаштування Браузера
8 </body>
9 </html>
10
```

Виконати Плагіни Вікно ?

- Виконати... F5
- Get PHP help Alt+F1
- Wikipedia Search Alt+F3
- Open file in another instance Alt+F6
- Змінити комб. клавіш/Видалити команду...

Запустити...

Введіть параметр запуску програми

...

Пуск! Зберегти... Скасувати

Открытие

« Chrome » Application

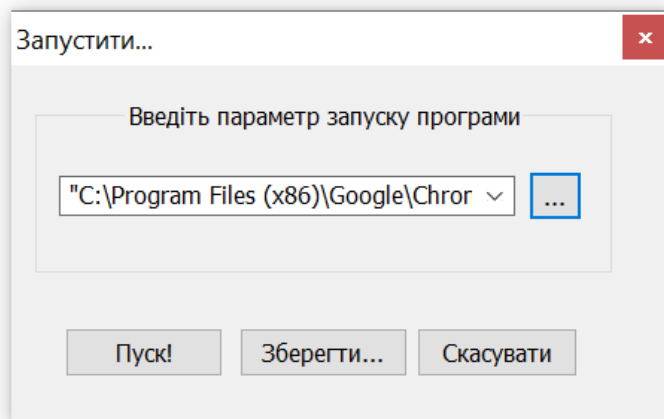
Поиск: Application

Упорядочить Новая папка

Имя	Дата изменения
88.0.4324.104	26.01.2021 7:07
SetupMetrics	29.01.2021 17:41
chrome.exe	21.01.2021 5:18
chrome_proxy.exe	21.01.2021 5:18

Имя файла: Executable file : (*.exe;*.com;*.c)

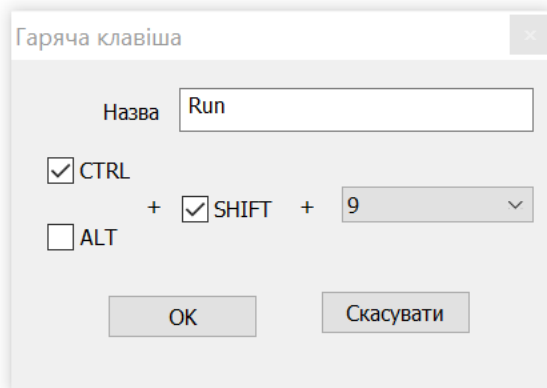
Открыть Отмена



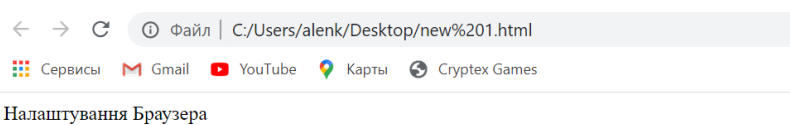
Але це тільки запуск браузера, а нам треба відкрити файл в цьому браузері, тому до наявному шляху потрібно дописати команду `$(FULL_CURRENT_PATH)`

```
"C:\Program Files (x86)\Google\Chrome\Application\chrome.exe" "$(FULL_CURRENT_PATH)"
```

Натискаємо Зберегти, після чого відкриється вікно в якому потрібно ввести назву рядка меню і задати гарячі клавіші.



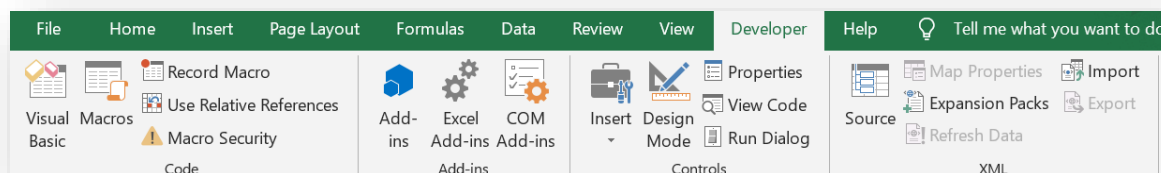
Запускаємо



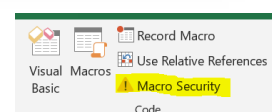
ДОДАТОК Б

Робота з VB в середовищі MS Office

На початку роботи в офісному додатку необхідно включити вкладку/**Ribbon Developer tab** «Розробник» (Option – Customize Ribbon – Deleloper)

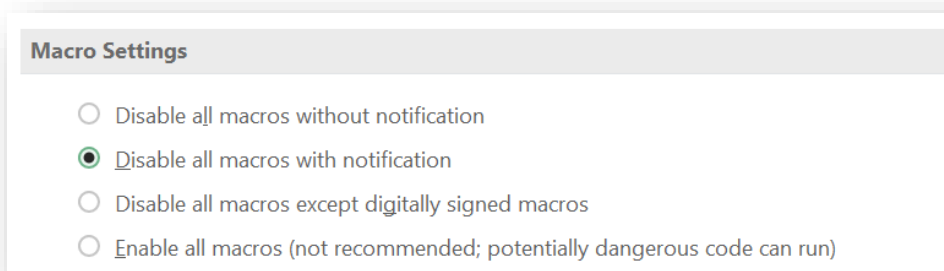


Після того, як вкладка Розробника буде активована, необхідно включити макроси. На вкладці Розробник знаходимо Безпеку макросів



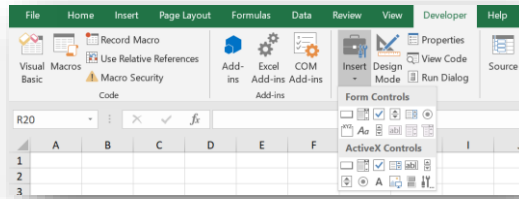
У вікні з налаштуваннями можна активувати відразу все макроси. Однак, слід звернути увагу на те, що розробники Microsoft не рекомендують вибирати цей варіант, так як є ймовірність запуску небезпечної програми, що може нашкодити комп'ютеру. Тому, виконуючи цю операцію, пам'ятайте, що ви дієте на свій страх і ризик. Рекомендується 2-й пункт 😊

9. **Вимкнути всі макроси без сповіщення:** вимикає всі макроси в документах та всі попередження про безпеку, пов'язані з макросами. Ця опція полягає у виборі, чи не довіряєте ніякому макросу.
10. **Вимкнути всі макроси за допомогою сповіщення:** це налаштування за замовчуванням. Це дозволяє активувати макроси у кожному конкретному випадку. Макроси вимкнені, але попередження про безпеку з'являються у присутності макросу.
11. **Вимкнути всі макроси, за виключенням цифрових підписаних макросів:** так само, як і в налаштуваннях за умовчанням, ця опція все ще дозволяє цифровим підписаним макросам, надісланим електронною поштою для запуску. Якщо це ще не так, ви отримаєте сповіщення, що дасть вам можливість це зробити;
12. **Увімкнути всі макроси:** цей параметр сильно не рекомендується, оскільки він показує комп'ютер, що виконує шкідливий код.

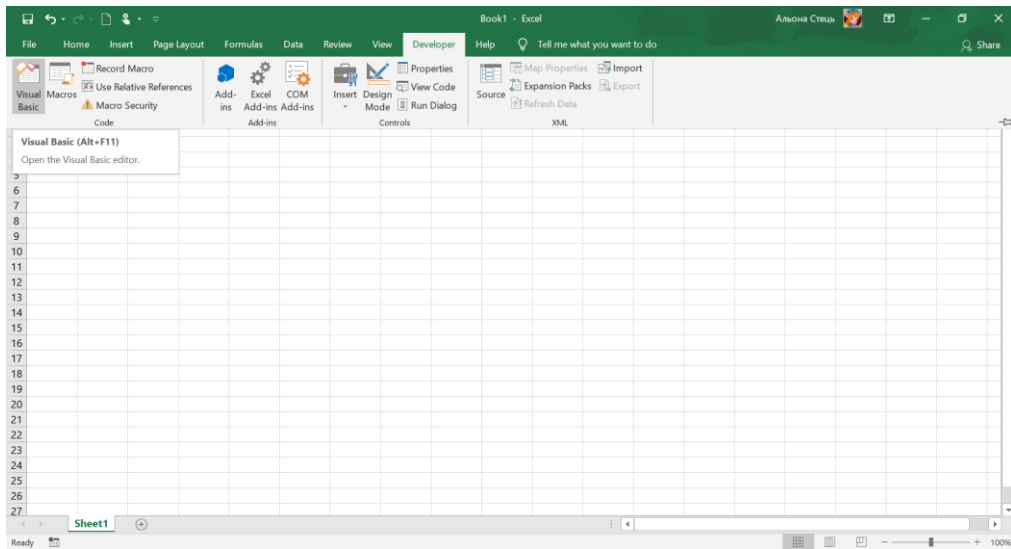


Далі можна розпочинати роботу з макросами.

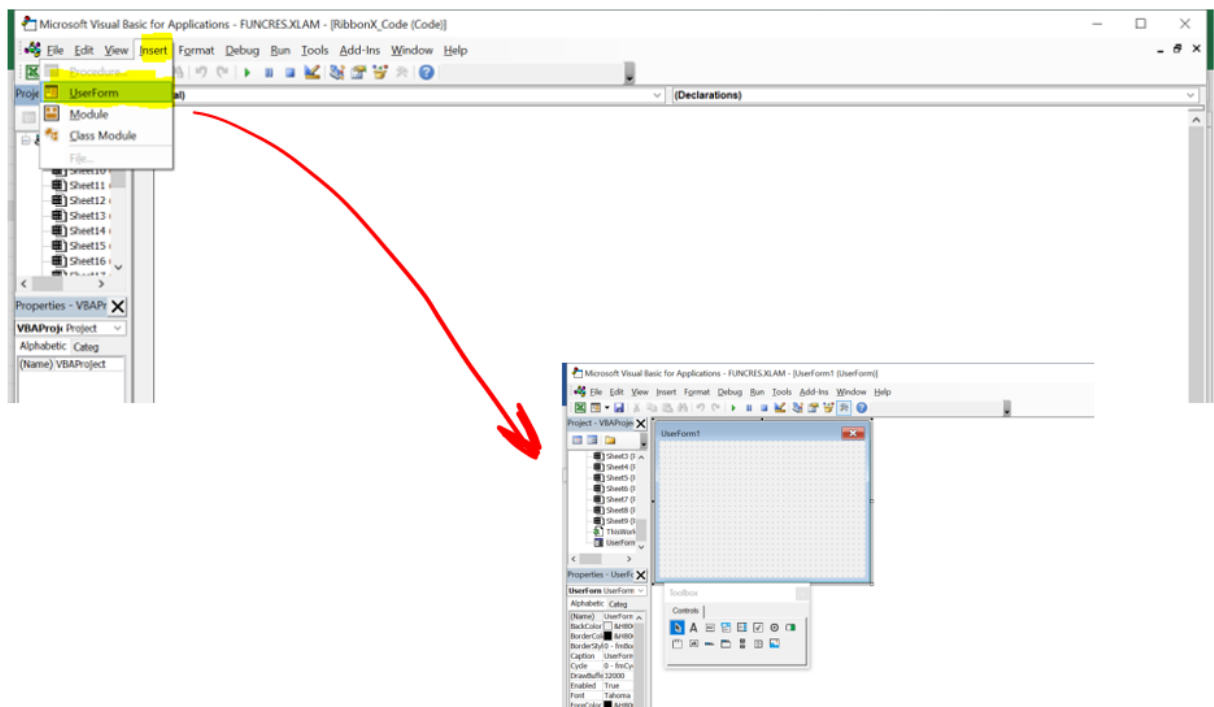
Всі інструменти: кнопки, режим конструктора, в якому можна змінювати властивості, переглядати код тощо, знаходяться на вкладці Розробник



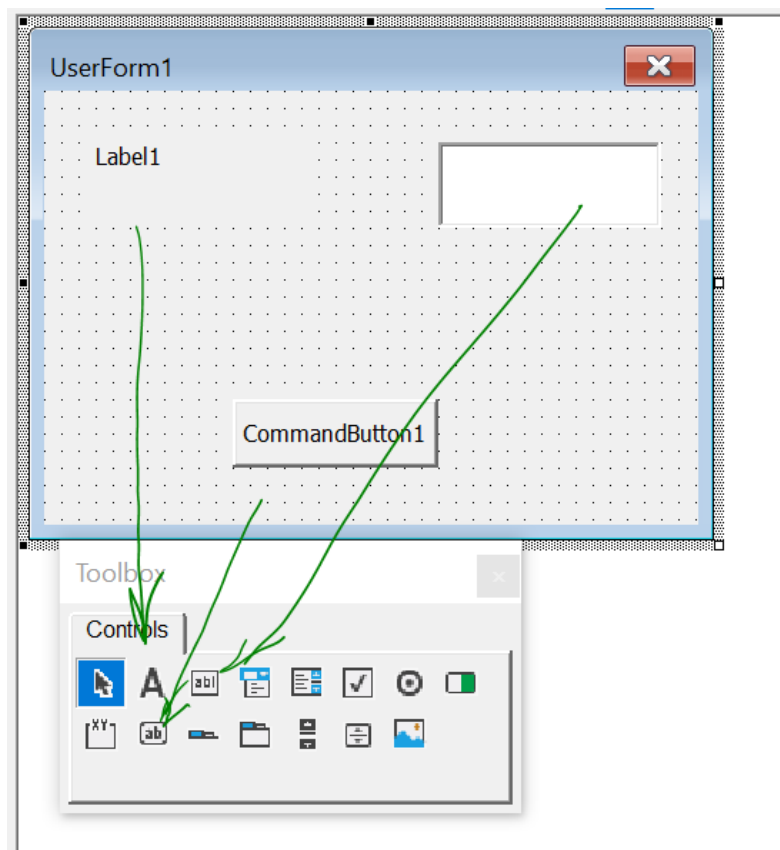
Відкриваємо VB :



Створюємо форму:

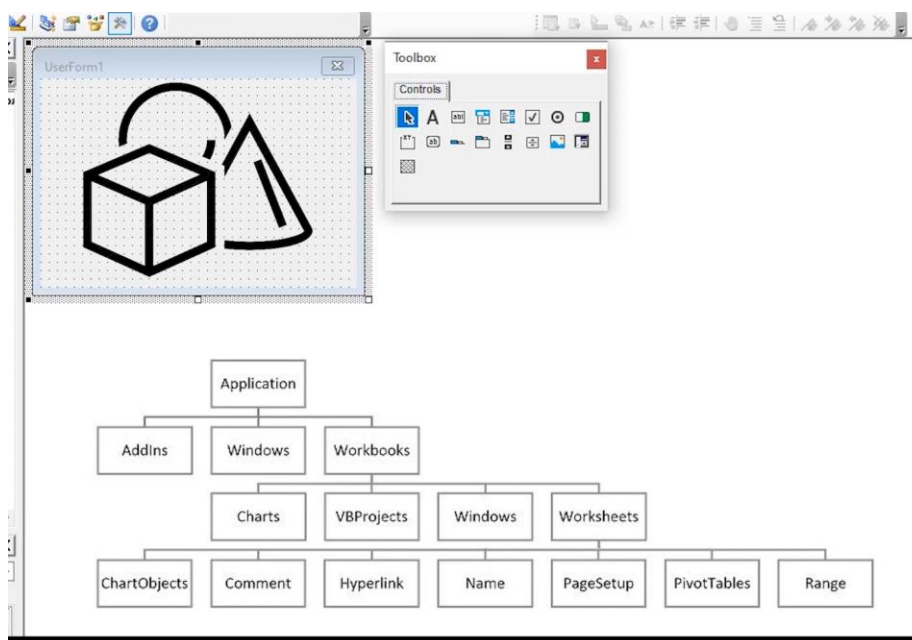


Розміщуємо елементи на форму:



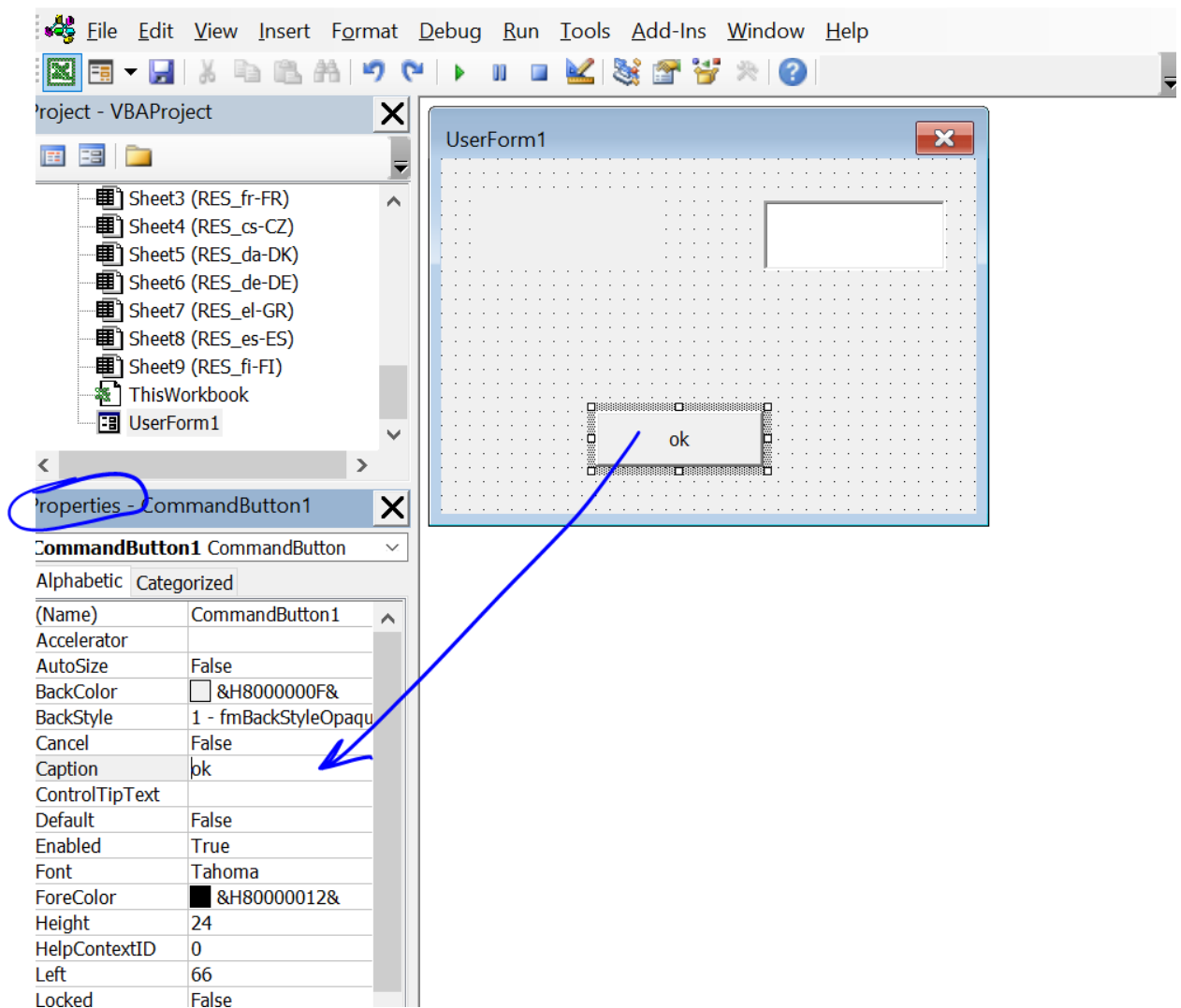
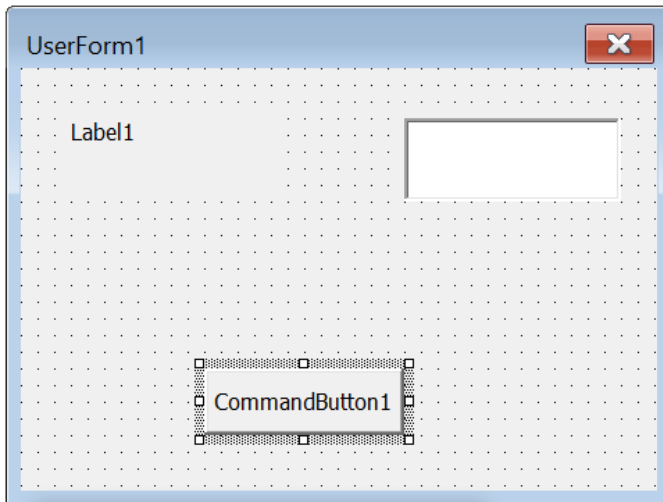
!!! Зверніть увагу на те, як саме називаються відповідні елементи (саме з такими назвами вони мають фігурувати в коді)

Нагадаємо, що до початку роботи всі об'єкти мають бути створеними та існувати в пам'яті комп'ютера



Приклад:

Ввести число а, обчислити формулу та вивести результат 😊



Project - VBAProject

- Sheet3 (RES_fr-FR)
- Sheet4 (RES_cs-CZ)
- Sheet5 (RES_da-DK)
- Sheet6 (RES_de-DE)
- Sheet7 (RES_el-GR)
- Sheet8 (RES_es-ES)
- Sheet9 (RES_fi-FI)
- ThisWorkbook
- UserForm1

Properties - CommandButton1

CommandButton1 CommandButton	
Alphabetic Categorized	
(Name)	CommandButton1
Accelerator	
AutoSize	False
BackColor	&H8000000F&
BackStyle	1 - fmBackStyleOpagu
Cancel	False
Caption	ok
ControlTipText	
Default	False
Enabled	True
Font	Tahoma
ForeColor	&H80000012&
Height	24
HelpContextID	0
Left	66
Locked	False

```
UserForm

Private Sub CommandButton1_Click()

Dim S As Double
Label1.Caption = "Введіть число"
a = Val(TextBox1.Text)
S = Sqr(a) / (4 * a ^ 2)
MsgBox " S=" & Str(S)

End Sub

Private Sub UserForm_Click()

End Sub
```

Код:

```
Private Sub CommandButton1_Click()
```

```
Dim S As Double
```

```
Label1.Caption="Введіть число"
```

```
a = val(TextBox1.Text)
```

```
S = Sqr(a) / (4 * a ^ 2)
```

```
MsgBox " S=" & Str(S)
```

```
End Sub
```

```
Private Sub UserForm_Click()
```

```
End Sub
```

Запускаємо!

