

## Створення тестового проекту засобами програми ділової графіки Power Point із використанням VBA.

У наш час існує велика кількість програмних засобів, призначених для перевірки знань учнів та генерування навчальних завдань. Але, незважаючи на це, у вчителя може виникнути потреба створити власну тестуючу програму.

Для розв'язання такої задачі на сьогоднішній день існує чимало середовищ розробки (Delphi, Visual C++, Visual Basic і т.д.), які дозволяють ефективно створювати будь-які програмні засоби. Разом із тим сучасні візуальні середовища потребують окремого вивчення і деякі специфічні можливості (анімаційні, звукові ефекти, нестандартне оформлення інтерфейсу програми) реалізувати в них не так просто.

Тому мета даної статті – познайомити читачів, перш за все тих, що не мають досвіду роботи з візуальними середовищами, з реалізацією тестового проекту засобами програми ділової графіки PowerPoint із використанням VBA.

Відомо, що основна ідея тестуючих програм – надати можливість користувачу дати відповідь на деяке питання або розв'язати певну задачу. Найпоширенішими є такі типи тестових завдань:

- а) вибір правильної відповіді з декількох запропонованих;
- б) вибір декількох правильних відповідей з множини запропонованих;
- в) установлення відповідності між елементами двох множин;
- г) визначення правильної послідовності дій.

Використання різних типів тестових завдань, надає можливість вчителю більш ретельно перевіряти рівень знань учня, уникати ситуацій, у яких учні могли б вгадати відповідь. Крім того, існує навчальний матеріал, перевірка знань по якому, потребує застосування різних типів тестових завдань, які б містили не лише питання з можливістю вибору правильної відповіді чи однозначної відповіді ("Так", "Ні").

Найпростіше у PowerPoint створити тестуючу програму з питаннями типу а).

Далі розглянуто методика створення такої програми.

Щоб розуміти принципи функціонування VBA-коду, необхідно мати деякі уявлення про структуру VBA-проекту.

VBA-проект складається з наступних структурних елементів:

1. Microsoft PowerPoint Об'єкти (Object).
2. Форми (Forms).
3. Модулі (Modules).
4. Модулі класу (Modules Classes).

Дерево структурних елементів можна побачити у вікні "Проек-VBA Project", яке активізується за допомогою команди "Вид-Вікно проекту" (View-Project Explorer).

Структурний елемент "Модулі класу" у даному випадку не використовуватиметься, тому нижче розглянуто призначення перших трьох структурних елементів.

Після того, як з використанням пункту меню "Сервіс" головного меню Power Point створюється новий макрос, VBA відразу створює модуль, який по замовчанню іменується Модуль1 (Module1), у якому відображається заголовок та текст макросу. Отже основне призначення модуля – збереження певних даних і текстів макросів, що обробляють ці дані. Варто зауважити, що поняття "Макрос" для VBA аналогічне до поняття "процедура" для VB. Щоб створити порожній модуль, потрібно з головного меню редактора VBA-проекту вибрати команду "Вставка-Модуль" ("Insert-Module").

Як відомо форми у Windows використовуються для відображення деякої інформації та для приймання від користувача певних даних для їх, подальшої обробки. Форми VBA у цьому відношенні не є винятком.

Структурний елемент "Microsoft PowerPoint Об'єкти" використовується для відображення даних про елементи керування (об'єкти), які розташовує на слайді користувач.

Якщо користувач не створював макроси, форми або не розташовував на слайді елементи керування, то відповідні структурні елементи не будуть відображатись у вікні "Проек-VBA Project".

Загальна структура тестуючої презентації з вибором однієї правильної відповіді з декількох запропонованих має бути такою.

Перший слайд – титульний. На інших слайдах, крім останнього, розташовані питання тесту: по одному питанню на кожному слайді. Напроти кожного варіанту відповіді на питання тесту розташована стандартна кнопка PowerPoint. Після активізації кнопки здійснюється перехід до наступного слайда. Якщо користувач вибрав кнопку, що відповідає правильній відповіді, то він отримує додатковий бал. На останньому слайді має бути розташована кнопка "Інформація". При активізації цієї кнопки з'являється форма VBA, що містить відомості про оцінку за тестування.

Практично все вище перераховане реалізується так.

Вважається, що на усіх слайдах розташовані необхідні об'єкти. За кнопками, які відповідають неправильним відповідям закріплено посилання на наступний слайд. За кнопками, які відповідають правильним відповідям закріплено макрос, який називається TrueAnswer. Крім того, потрібна змінна, у яку буде записуватись кількість правильних відповідей. Код, який міститиме опис змінної та макрос, може мати такий вигляд:

```
Public k As Integer
```

```
Sub TrueAnswer()
```

```
    k=k+1
```

```
    ActivePresentation.SlideShowSetting.Run
```

```
    ActivePresentation.SlideShowWindow.View.Next
```

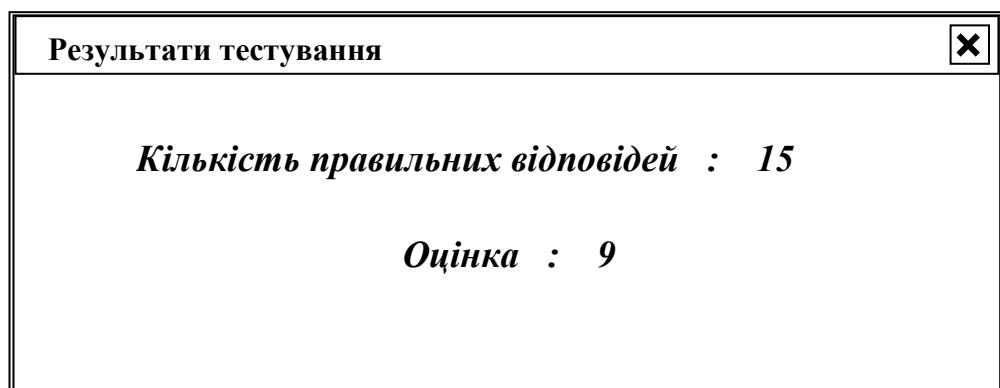
```
End Sub
```

Зарезервоване слово Public вказує на те, що змінна k буде доступною у будь-якому місці VBA-проекту. Два останні рядки потрібні для здійснення переходу на наступний слайд.

Щоб створити макрос TrueAnswer потрібно:

1. Викликати редактор VBA-проекту, користуючись командою "Сервіс-Макрос-Редактор Visual Basic".
2. Виконати команду "Вставка-Модуль" редактора VBA-проекту.
3. У вікні "[ім'я\_презентації]-Модуль1(Програма)" набрати вище вказаний код макросу та рядок опису змінної k.

Для перегляду результатів тестування, за кнопкою "Інформація" останнього слайду потрібно закріпити макрос ShowResult. Цей макрос має відображати приблизно таку форму.



Результати тестування

*Кількість правильних відповідей : 15*

*Оцінка : 9*

Щоб створити форму потрібно у вікні редактора VBA-проекту виконати команду "Вставка-UserForm". Після цього, користуючись панеллю "Панель елементів", потрібно

розташувати на формі чотири елементи керування "Напис". Використавши вікно "Властивості-UserForm1" змінити властивість "Caption" перших двох елементів відповідно на "**Кількість правильних відповідей** :'" та "**Оцінка** :'". Властивість "Caption" решти написів не змінювати.

Якщо у VBA-проекті деякі панелі не відображені, то для їх, відображення потрібно використати пункт меню "Вид" (View) головного меню редактора VBA-проекту:

- "Вид-Вікно властивостей" (View-Properties Window),
- "Вид-Панель елементів" (View-Toolbox).

Після налаштування зовнішнього вигляду форми у модулі "Модуль1" потрібно створити макрос ShowResult, який міститиме приблизно такий код.

```
Sub ShowResult()  
Dim bal As double  
    ' обробка змінної k для отримання оцінки за тестування  
    ...           ...           ...  
  
    ' bal – змінна, що містить оцінку за тестування  
  
    Label3.Caption=k  
    Label4.Caption=bal  
  
    UserForm1.Show  
    k=0  
    bal=0
```

End Sub

Зауваження. 1. Символ "" використовується у VB для відображення у тексті коду проекту коментарів. 2. Рядок "Dim bal As double" дозволяє описати локальну змінну дійсного типу.

Варто зауважити, що з формою можна працювати у двох режимах – режимі "об'єкт" та режимі "програма". Режим "об'єкт" дозволяє розташовувати на формі елементи керування (кнопки, списки, мітки і т. д.) та налаштовувати їх властивості (колір, розміри і т. д.). Режим "програма" надає можливість у вигляді процедур оформити обробники подій елементів керування розташованих на формі, та самої форми.

Під подією розуміють певну дію, яку виконав користувач над елементом керування (клацання чи рух курсору миші по елементу керування, натиснення клавіш на клавіатурі і т. д.). Крім того до подій відносять певні дії операційної системи (завантаження форми, руйнування форми, подія таймера).

Щоб перейти від одного режиму форми до іншого можна скористатися відповідними кнопками, розташованими у верхньому правому кутку вікна "Проек-VBA Проект".

Для обробки змінної k можна використати оператор розгалуження, який, в залежності від кількості правильних відповідей, дозволить вирахувати оцінку за тестування, наприклад, по дванадцятибальній шкалі.

У Visual Basic цей оператор має таку структуру.

```
If <умова> Then  
    <група операторів 1>  
Else  
    <група операторів 2>           повне розгалуження.  
End If
```

```
If <умова> Then  
    <група операторів >           неповне розгалуження.  
End If
```

Прості умови формуються у вигляді змінних чи констант, розділених одним з операторів порівняння. Складені умови утворюються завдяки використанню логічних операцій OR та AND. Наприклад:

```

If a>5 and b=3 then
    c=c+5 : d=d+1
Else
    c=c-3
End if

```

Важливо пам'ятати, що у бейсіку під кожний оператор формально відводиться один рядок. Якщо ж необхідно у одному рядку записати декілька операторів, то їх потрібно розділяти символом ":", як показано у прикладі. Крім того оператор присвоєння та операція порівняння позначаються одним і тим же символом "=".

Звичайно, якщо кожне питання у тесті оцінюється однаково, то можна обійтись і без використання оператора розгалуження, скориставшись залежністю  $bal=k*12/n$  (k–кількість правильних відповідей, n – кількість питань у тесті).

Нижче описано методику створення слайдів тестуючої презентації з використанням питання типу "вибір кількох правильних відповідей з кількох запропонованих".

Тут, для реалізації таких питань, пропонується затосовувати елемент керування "вимикач" (ToggleButton), хоча простіше було б скористатись елементом керування "прапорець" (CheckBox), але на слайді цей елемент має досить непривабливий вигляд.

Структура презентації може бути такою.

Перший слайд – титульний. На інших слайдах, крім останнього, розташовані питання тесту: по одному питанню на кожному слайді. Напроти кожного варіанту відповіді на питання тесту розташований елемент керування "вимикач", на якому зображено напис "Ні"(тобто вимикач вимкнено) або "Так" (тобто вимикач увімкнено). Після вибору вимикача мишею напис на ньому змінюється з "Ні" на "Так". Увімкнувши вимикачі в положення, що відповідають правильній відповіді, користувач здійснює перехід до наступного питання тесту, натиснувши стандартну кнопку Power Point "Далі".

Якщо користувач вибрав положення вимикачів, що відповідають правильній відповіді, то він отримує додатковий бал. На останньому слайді має відобразитись інформація про кількість правильних відповідей та оцінку за тестування.

Практично все вище перераховане реалізується так.

Будемо вважати, що на усіх слайдах розташовані необхідні об'єкти. За стандартними кнопками Power Point "Далі", за винятком першої та останньої, закріплено макроси з іменами TrueAnswerN(N+1), де N – номер слайду. За кнопкою "Далі" першого слайду закріплено макрос Start. За кнопкою "Далі" останнього слайду закріплено макрос Finish.

Для усіх вимикачів на кожному слайді, властивість Value повинна мати значення False, властивість Caption– значення "Ні".

На останньому слайді розташовані чотири елемента керування "напис".Перші два з них мають властивість Caption, відповідно "Кількість правильних відповідей" та "Оцінка за тестування".

Макрос Start може бути описаний у модулі проекту. Його призначення– обнулення лічильника кількості правильних відповідей та надання властивості Value вимикачів слайду з першим запитанням тесту значення False. Текст цього макросу може бути, зокрема, такий.

```

Public k As integer
Sub Start()
    k=0
    Slide2.ToggleButton1.Value=False
    Slide2.ToggleButton2.Value=False
    Slide2.ToggleButton3.Value=False

```

'здійснюємо перехід до наступного слайду

```

ActivePresentation.SlideShowSettings.Run
ActivePresentation.SlideShowWindow.View.Next

```

```

End Sub

```

ToggelButtonN (N – певне натуральне число) – це ім'я, яке VBA по замовчанню надає елементу керування "вимикач". Slide2 – це назва слайду, на якому розташовані вимикачі. Необхідно пам'ятати (див. вище), що при розташуванні на слайді будь-якого елемента керування VBA до проекту автоматично додається у структуру "Microsoft PowerPoint Об'єкти (Object)" об'єкт з ім'ям цього слайду. Він надає можливість користувачу змінювати властивості елементів керування та створювати обробники подій цих елементів.

Нижче описано макрос, що аналізує правильність відповіді при переході від N-того до N+1-шого слайду. Для прикладу розглянуто перехід від другого до третього слайда. Вважатимемо, що на другому слайді три варіанти відповіді, (причому відповідь вважається правильною при вмиканні першого та третього вимикача) на третьому – шість.

```
Sub TrueAnswer23()
```

```
    'аналізуємо правильність відповіді на питання, розташоване  
    'на другому слайді
```

```
    if Slide2.ToggelButton1.Value and not(Slide2.ToggelButton2.Value) _  
       Slide2.ToggelButton3.Value then k=k+1
```

```
    'встановлюємо значення властивості Value усіх вимикачів  
    'на третьому слайді у False
```

```
    Slide3.ToggelButton1.Value=False  
    Slide3.ToggelButton2.Value=False  
    Slide3.ToggelButton3.Value=False  
    Slide3.ToggelButton4.Value=False  
    Slide3.ToggelButton5.Value=False  
    Slide2.ToggelButton6.Value=False
```

```
    'здійснюємо перехід до наступного слайда
```

```
    ActivePresentation.SlideShowSettings.Run  
    ActivePresentation.SlideShowWindow.View.Next
```

```
End Sub
```

Зауваження. Символ "\_" використовується для здійснення розриву рядка з командою VB, яка є занадто довгою.

Макрос Finish може містити приблизно такий код.

```
Sub Finish()
```

```
    Dim bal As integer
```

```
    'F – номер останнього слайду
```

```
    bal=k*12/n ' n – кількість питань у тесті  
    SlideF.Label3.Caption=i 'кі-сть правильних відповідей  
    SlideF.Label4.Caption=bal 'оцінка за тестування
```

```
    'здійснюємо перехід до наступного слайду (останнього)
```

```
    ActivePresentation.SlideShowSettings.Run  
    ActivePresentation.SlideShowWindow.View.Next
```

```
End Sub
```

Крім вище описаних макросів потрібно також описати обробник події "клацання лівою клавішею миші на вимикачі". Цей обробник потрібний для покращення візуального сприймання стану вимикача – "увімкнено" чи "вимкнено". Для створення такого обробника потрібно двічі клацнути лівою клавішею миші на елементі керування

"вимикач", розташованому на слайді, у режимі "Слайд", та у вікні редактора проекту [Ім'я\_презентації]-SlideN(програма) набрати такий код.

```
Private Sub ToggleButton1_Click()  
  
    If ToggleButton1.Value Then  
        ToggleButton1.Caption="Так"  
    Else  
        ToggleButton1.Caption="Ні"  
    End If
```

```
End Sub
```

Необхідно зауважити, що подібний обробник потрібно створити для кожного елемента керування "вимикач".

Далі описано методику створення тестуючої програми з використанням типу питань "встановлення відповідності між елементами двох множин". Для реалізації такої програми застосовується елемент керування "Список, що випадає" (ComboBox).

Загальна структура програми-презентації має бути такою.

Перший слайд – титульний. На інших слайдах, крім останнього, розташовані питання тесту: по одному питанню на кожному слайді. Питання формуються у такому вигляді. На слайді розташовується набір об'єктів напроти кожного з яких знаходиться елемент керування "список, що випадає". Відповідь на питання полягає у правильному виборі записів із списку. Наприклад на слайді розташовано три геометричні тіла: конус, призма та циліндр. Завдання учня, що тестується, правильно обрати назви цих тіл у відповідних списках. У такому випадку встановлюється однозначна відповідність між геометричними тілами та їх назвами.

На останньому слайді має відобразитись інформація про кількість правильних відповідей та оцінку за тестування.

Практично все вище перераховане реалізується так.

Будемо вважати, що на усіх слайдах розташовані необхідні об'єкти. На першому (титульному) слайді повинна бути присутня кнопка "Далі". За цією кнопкою потрібно закріпити макрос GoFrom1to2, який міститиме, наприклад, такий код.

```
Public Sub GoFrom1to2()  
    k = 0  
  
    Slide2.ComboBox1.Clear  
    Slide2.ComboBox2.Clear  
    Slide2.ComboBox3.Clear  
  
    Slide2.ComboBox1.AddItem ("Піраміда")  
    Slide2.ComboBox1.AddItem ("Конус")  
    Slide2.ComboBox1.AddItem ("Призма")  
    Slide2.ComboBox1.AddItem ("Циліндр")  
    Slide2.ComboBox1.AddItem ("Сфера")  
    Slide2.ComboBox1.AddItem ("Куб")  
    Slide2.ComboBox2.List = Slide2.ComboBox1.List  
    Slide2.ComboBox3.List = Slide2.ComboBox1.List  
  
    ActivePresentation.SlideShowSettings.Run  
    ActivePresentation.SlideShowWindow.View.Next
```

```
End Sub
```

Тут k - глобальна змінна, описана у деякому модулі VBA-проекту. Сам макрос потрібно описати також у цьому модулі.

Команда Slide2.ComboBox1.Clear дозволяє видалити вміст списку ComboBox1, що розташований на другому слайді.

Команда `Slide2.ComboBox1.AddItem ("Піраміда")` здійснює додавання у кінець списку `ComboBox1` запису "Піраміда". Якщо список порожній, то запис додається у першу позицію. Важливо пам'ятати, що індексація записів списку починається з нуля.

Команда `Slide2.ComboBox2.List = Slide2.ComboBox1.List` переписує вміст списку `ComboBox1` у список `ComboBox2`.

На другому слайді за кнопкою "Далі" потрібно закріпити макрос `GoFrom2to3()`, який дозволяє оцінити правильність відповіді на питання, розташоване на другому слайді, заповнює даними списки, розташовані на третьому слайді та здійснює перехід до наступного слайду. Текст макросу може бути, зокрема, таким.

```
Public Sub GoFrom2to3()
```

```
If Slide2.ComboBox1.Text = "Конус" Then k = k + 1  
If Slide2.ComboBox2.Text = "Цидіндр" Then k = k + 1  
If Slide2.ComboBox3.Text = "Призма" Then k = k + 1
```

```
Slide3.ComboBox1.Clear  
Slide3.ComboBox2.Clear  
Slide3.ComboBox3.Clear
```

```
Slide3.ComboBox1.AddItem ("Основа")  
Slide3.ComboBox1.AddItem ("Перпендикуляр")  
Slide3.ComboBox1.AddItem ("Трьохгранний кут")  
Slide3.ComboBox1.AddItem ("Двогранний кут")  
Slide3.ComboBox1.AddItem ("Октаедр")  
Slide3.ComboBox1.AddItem ("Дотична площина")
```

```
Slide3.ComboBox2.List = Slide3.ComboBox1.List  
Slide3.ComboBox3.List = Slide3.ComboBox1.List
```

```
ActivePresentation.SlideShowSettings.Run  
ActivePresentation.SlideShowWindow.View.Next
```

```
End Sub
```

Варто зауважити, що на відміну від VB у VBA елемент керування "список, що випадає" не має властивості "list", яка дозволяє заповнити список даними на етапі розробки проекту, тому це доводиться виконувати із VBA-коду, як показано вище.

Команда `Slide2.ComboBox1.Text` дозволяє звернутися до запису, розташованому у віконці списку, що випадає.

У даному випадку за кожний правильно обраний запис у списку нараховується додатковий бал.

Макроси для інших слайдів створюються цілком аналогічно.

За кнопкою "Далі" передостаннього слайду має бути закріплений макрос, що ініціалізує змінні, які містять дані про кількість правильних відповідей та оцінку за тестування. Методика створення такого макросу описана вище.

Для створення питань типу "визначення правильної послідовності дій" також можна використати елемент керування "список, що випадає".

Щодо практичної реалізації вище описаних тестуючих презентацій варто зауважити наступне:

1. Перш ніж розташовувати на слайдах елементи керування, потрібно детально розробити структуру презентації та створити усі необхідні слайди, оскільки після зміни порядку розташування слайдів може виникнути плутанина при зверненні до властивостей елементів керування, особливо у випадку, коли елементи керування мають імена, прийняті по замовчанню.

2. При використанні елементів керування VBA, корисно давати їм більш осмислені імена, ніж ті, що пропонуються по замовчанню. Для цього потрібно змінювати властивість "name" елемента, користуючись вікном "Властивості" ("Properties").
3. Використовуйте властивості "BackColor", "Font", "Picture" для налаштування зовнішнього вигляду елемента керування.
4. У випадку, коли макрос працює не коректно або взагалі не працює, використовуйте для відлагодження макросу кнопки "Завантажити", "Пауза" та "Стоп" ("Run Macro", "Break" та "Reset"), які розташовані на стандартній панелі вікна редагування VBA-проекту.
5. Необхідно пам'ятати, що при використанні точок переривань (Break Point) у вікні VBA-проекту, під час виконання презентації, Power Point не здійснює зупинок на цих точках, тобто відлагодити макрос можна лише під час розробки макросу у вікні VBA-проекту, але не на етапі виконання презентації.

Вище описано методики створення тестуючих проектів, які містять питання певного типу. Зрозуміло, що у більшості випадків, виникає потреба у одному тестовому завданні використовувати різні типи питань. Цього можна досягти, комбінуючи описані способи створення тестуючих програм.

Даний матеріал може бути використаний вчителем інформатики, як пропедевтика вивчення візуального середовища розробки Visual Basic на факультативних заняттях. Крім того, познайомивши учнів із використанням VBA у Power Point, вчитель може, у подальшому, вивчати з ними застосування VBA під Word та Excel.

У профільних класах, де інформатика вивчається починаючи із сьомого класу, учні старших класів виконують курсові проекти. Як правило, основною тематикою створення таких проектів є створення саме тестуючих та навчальних програм, оскільки у сучасних школах вчителі з різних предметів активно використовують нові інформаційні технології для повідомлення нового матеріалу, перевірки та корекції знань учнів. Одним із засобів для створення подібних проектів може бути Power Point.

Використана література:

1. Під ред. Ю. І. Машбиця. Основи нових інформаційних технологій навчання. – К.: ІЗМН, 1997. – 264 с.
2. Брайан Сайлер, Джефф Споттс. Использование Visual Basic 6. – издательский дом "Вильямс", 2000. – 830 с.
3. Онищенко С. М. Основні можливості навчаючої програми-тренажера Arbeit// Комп'ютер у школі та сім'ї, №1, 2000, с 20-23.