

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКА НАЦІОНАЛЬНА АКАДЕМІЯ МІСЬКОГО
ГОСПОДАРСТВА

Б.І. Погребняк

МЕТОДИЧНІ ВКАЗІВКИ

**для виконання розрахунково-графічної роботи
з курсу**

"ЕКОНОМІЧНА ІНФОРМАТИКА"

*(для студентів 1 курсу денної форми навчання бакалаврів напрямку
6.030504 "Економіка підприємства" і 6.030509 "Облік і аудит")*

ХАРКІВ - ХНАМГ - 2009

Методичні вказівки для виконання розрахунково-графічної роботи з курсу "Економічна інформатика" (для студентів 1 курсу денної форми навчання бакалаврів по напрямку 6.030504 "Економіка підприємства" і 6.030509 "Облік і аудит") / Укл.: Погребняк Б.І. - Харків: ХНАМГ, 2009. – 20 с.

Укладач: Б.І. Погребняк

Рекомендовано кафедрою "Прикладна математика та інформаційних технологій", протокол № 9 від 24 березня 2009 р.

Вступ

Найперший варіант мови програмування BASIC (Бейсік) був створений у 1964 р. співробітниками Дартмутського коледжу Дж. Кемені й Т. Курцем. Історія походження назви мови дуже цікава. У минулому столітті один англійський місіонер виділив з англійської мови триста найбільш поширених слів, назвав їх Basic English і став навчати місцевих жителів. Досвід виявився успішним, контакти з ними значно спростилися. Творці мови BASIC прагнули досягти того ж ефекту – полегшити взаєморозуміння між починаючими програмістами і комп'ютером. Аббревіатура BASIC так і розшифровується – «Beginner's All purpose Symbolic Instruction Code», що в перекладі значить «багатоцільова мова символічних команд для початківців».

Мова програмування VBA – Visual Basic for Applications є результатом подальшого розвитку мови BASIC. Незважаючи на те, що VBA для кожного з додатків, що її підтримує (Microsoft Excel, Microsoft Word і т.д.), має свій, особливий набір можливостей, загальна основа в неї залишається незмінною. Тобто, освоївши один раз VBA, наприклад, для Microsoft Excel, використання її потім, наприклад, для автоматизації Microsoft Word не складе особливого клопоту.

Основні теоретичні положення

Будь-яка мова програмування складається із:

- 1) *інструкцій (операторів, команд, пропозицій і т.д.)* і
- 2) *коментарів.*

Інструкції служать для опису самого процесу обробки інформації і призначені для виконання комп'ютером. Коментарі ж після трансляції ніяких машинних команд не породжують. Вони призначені для пояснення найбільш складних місць програми й адресовані людині – тобто програми пишуть для комп'ютерів, а читають люди.

У будь-якій мові програмування є два типи інструкцій, що використовуються для опису:

- 1) *даних* – інформаційних об'єктів, що беруть участь у процесі обробки, і
- 2) *алгоритмів* – наборів розпоряджень, відповідно до яких ці дані обробляються.

Будь-яка програма, на будь-якій мові програмування, будь-якого ступеня складності складається з різних сполучень наступних 3-х базових алгоритмів:

- 1) *лінійного,*
- 2) *розгалуженого* і
- 3) *циклічного.*

У лінійному алгоритмі всі інструкції виконуються строго послідовно, одна за одною, від початку програми і до кінця. Алгоритм, що розгалужується, дозволяє виконати одну частину інструкцій і пропустити іншу. Циклічний обчислювальний процес служить для багаторазового повторення деякої послідовності інструкцій, що називається тілом циклу.

Одержати вихідний код на VBA можна двома різними способами:

- 1) *автоматично* – записуючи виконувані користувачем команди і трансформуючи їх в інструкції VBA, чи
- 2) *вручну* – вводячи необхідний текст із клавіатури.

Текст, створений першим способом, традиційно називається *макросом*, тоді як другим – *процедурою*. Після того як код програми був записаний, вже не можна визначити, як він туди потрапив – вручну чи автоматично.


Текст макросу (чи процедури) складається з інструкцій, що становлять:

- 1) *заголовок*,
- 2) *тіло*,
- 3) *закінчення*.

Заголовок визначає ім'я процедури і її початок у тексті модуля, а тіло – дані й виконувані над ними дії. Заголовок, у свою чергу, складається з:

- а) *ключового слова **Sub*** (від Subroutine – підпрограма),
- б) *імені процедури і*
- в) *пари круглих дужок – ()*.

Після тіла макросу завжди з'являється пара ключових слів **End Sub**, які говорять про те, що макрос закінчився.


 *Ключові (службові, зарезервовані) слова VBA призначені для внутрішніх потреб мови, їх не можна використовувати ні для яких інших цілей, крім – спеціально обговорених!*

Далі в тексті вони виділені **моноширинним напівжирним курсивом**. Тоді структура процедури у загальному видгляді виглядатиме так:

```
Sub<ім'я>( )  
    [<інструкції>]  
End Sub.
```

Коментар у мові програмування VBA позначається символом апострофа('). При цьому він може займати не тільки рядок повністю, але й частину – від місця його початку в рядку (знака апострофа) і до кінця.

Для створення, перегляду, редагування і запуску на виконання макросів служить інструмент, що називається Редактор Visual Basic. На виконання він запускається декількома різними способами. Найбільш прості з них – це:

- 1) за командою **Сервіс ⇒ Макрос ⇒ Редактор Visual Basic**,
- 2) за натисканням кнопки  **Редактор Visual Basic** на панелі інструментів **Visual Basic** (якщо вона, звичайно, активізована), чи
- 3) натиснувши сполучення клавіш **Alt + F11**.

Редактор Visual Basic багато в чому допомагає у процесі створення програм. Так, у першому рядку оголошення процедури необхідно ввести тільки ключове слова **Sub** і ім'я процедури, а коли курсор її залишить, автоматично будуть добавлені:

- 1) круглі дужки після імені процедури, і
- 2) рядок завершення процедури – **End Sub**.


Ключові слова, такі, наприклад, як **Sub**, при цьому також необов'язково вводити з великої букви – редактор Visual Basic це зробить сам. Навпаки, їх краще вводити малими літерами, оскільки якщо після того, як курсор залишить поточний рядок, ключові слова будуть виведені з великої букви, – це може служити додатковою ознакою того, що всі вони були введені правильно. Автоматичне додавання рядка завершення процедури **End Sub** також сприяє зменшенню імовірності появи в ній помилок – заголовок і завершення процедури сформовані, практично, автоматично – залишилося записати тільки її тіло?! Але і тут редактор Visual Basic допоможе – після введення, наприклад, ключового слова **MsgBox** і пробілу після нього з'являється допоміжне вікно, в якому відображається прототип функції. Ім'я аргументу, який тепер необхідно вводити, виділено в ньому **напівжирним** накресленням.

Якщо при роботі з макросами використовується кольоровий монітор, то редактор Visual Basic розцвічує їх у такий спосіб:

- 1) зеленим кольором виділяються **коментарі**,
- 2) синім кольором виділяються ключові слова, такі як **Sub, End Sub** та інші, і
- 3) чорним кольором – все інше.

Це служить додатковим засобом структурування тексту програми.

Після того, як створена нова, чи відредагована існуюча процедура, необхідно перевірити, як вона працює. Це можна виконати в такий спосіб:

1. Установити курсор у межах процедури, включаючи рядки її заголовка і завершення.
2. Запустити її на виконання одним із способів:
 - а) виконати команду **Run ⇨ Run Sub/UserForm**,
 - б) натиснути кнопку  **Run Sub/UserForm** на панелі інструментів **Standard**,
 - в) натиснувши клавішу **F5**.

Якщо у момент запуску процедури курсор знаходиться в поза її межами, Visual Basic не зможе визначити, яку саме процедуру необхідно запускати, і він виводить вікно діалогу **Macros**, за допомогою якого можна запустити будь-яку процедуру – і створену вручну, і записану автоматично.

Змінні у програмуванні аналогічні змінним, відомим усім зі шкільного курсу алгебри – ми пишемо ім'я змінної замість її значення, що на момент її *згадування* може бути ще невідомо, але яке буде відомо на момент її *використання*. Так само, коли в тексті програми зустрічається ім'я змінної, Visual Basic вставляє у відповідне місце її значення. Наприклад, якщо змінна `Var1` має значення 1, то весь вираз

$$\text{Var1}+1$$


матиме значення 2. При цьому, коли Visual Basic обчислює цей вираз, число 1, записане в змінну `Var1`, підставляється у вираз і складається з числом 1, безпосередньо зазначеним у ньому, в підсумку виходить 2. Якби в змінній `Var1` було записане число 5, увесь вираз одержав би значення 6.

Фізично змінна являє собою деяку ділянку оперативної пам'яті комп'ютера – комірку (клітинку), в якій записане *значення* якого-небудь *типу даних* для його подальшого використання. Розмір і внутрішня структура такої ділянки пам'яті залежать від типу даних, для збереження яких він призначений. Доступ до нього здійснюється за *іменем змінної*. Вміст ділянки пам'яті (значення змінної) у процесі виконання програми може змінюватися, а от ім'я і тип – ні.

У деякому сенсі змінна подібна клітинці Microsoft Excel – їй можна призначити ім'я і потім звертатися до неї з цим іменем, не піклуючись більше про те, які номери стовпця і рядка вона має. Точно так само ім'я змінної дозволяє звертатися до даних, що зберігаються в оперативній пам'яті комп'ютера, не думаючи потім, де саме вони розташовані – Visual Basic бере на себе всі турботи щодо забезпечення доступу до необхідних даних.

Більш формально – усяка змінна в будь-якій мові програмування має наступні три характеристики:

- 1) ім'я,
- 2) тип,
- 3) значення.

 У продовж виконання програми ім'я і тип змінної не змінюються, а от значення – може. Хоча, у Visual Basic змінна типу **Variant** у різні моменти часу може зберігати різні типи даних.

Будь-яка змінна VBA може бути створена одним з двох різних способів:

- 1) неявно – тобто просто вказавши її ім'я у тексті програми і
- 2) явно – за допомогою інструкції **Dim**.

При неявному способі Visual Basic створює змінну, і виділяє для неї пам'ять, як тільки знайде її ім'я в тексті програми. Звичайно це відбувається при записі в змінну деякого значення. Запис значення в змінну називається *присвоюванням значення*. Присвоювання значення змінної виконують за допомогою *оператора присвоювання*, що виглядає як знак рівності (=).

Наприклад, в інструкції присвоювання

$$\text{Var1} = 1$$

змінній Var1 присвоюється значення 1. При цьому виконуються різні процеси, залежно від того, чи зустрівся ім'я змінної у програмі:

- в перше, чи
- ні.

Якщо це перше згадування імені змінної у програмі, то Visual Basic:

- 1) резервує під неї ділянку пам'яті необхідного розміру,
- 2) записує в неї значення, наприклад, число 1, а також
- 3) присвоює виділеній ділянці пам'яті ім'я створеної в такий спосіб змінної.

Якщо ж така змінна вже існувала, то в результаті виконання оператора присвоювання в ділянку пам'яті, на яку вказує її ім'я, просто записується нове значення, наприклад, число 1. *У результаті колишній вміст цієї ділянки пам'яті (чи значення змінної) безповоротно втрачається.* Неявне створення змінних називають також створенням змінних на льоту (on-the-fly declaration). Усі змінні, створені в такий спосіб, одержують тип **VARIANT**.

Для того щоб явно створити змінну, використовують інструкцію **Dim** (скорочення від dimension – розмірність) з наступним синтаксисом:

```
Dim <ім'я> [As <тип>] [ , <ім'я> [As <тип>] ] . . . ,
```

де: ім'я – назва (ім'я) створюваної змінної, а

тип – тип створюваної змінної.

Ім'ям змінної в мові програмування Visual Basic може служити будь-як припустимий ідентифікатор. Ідентифікатор – це послідовність символів, що включає в себе букви, цифри і знак підкреслення (_). При цьому в якості імен змінних можуть використовуватися ідентифікатори, що формуються на основі таких правил:

- 1) Ім'я змінної повинне починатися з букви.
- 2) За першою буквою можуть йти різні комбінації букв, цифр і символів підкреслення.
- 3) Довжина імені змінної не може перевищувати 255 символів.
- 4) Ім'я змінної не повинне збігатися з жодним з ключових слів мови програмування.
- 5) Ім'я змінної повинне бути унікальним у межах її області видимості.

Наприклад, такі імена змінних є припустимими:

Var1 ,
BookName ,
Sheet_Name ,

а такі – ні:

1Var , 'Починається з цифри
Sheet Name 'Містить пробіл
Sub 'Збігається з ключовим словом.

Константи – це елементи внутрішніх дані програми, що протягом її виконання свого значення не змінюють, тобто у виразах вони можуть використовуватися тільки праворуч від оператора присвоєння (=). Константи, також як і змінні, займають деякі ділянки оперативної пам'яті комп'ютера (клітинку, комірку), внутрішня структура яких (чи внутрішнє представлення) також залежить від їхнього типу. Вони, також як і змінні, характеризуються:

- 1) типом і
- 2) значенням,
- а деякі з них – і
- 3) ім'ям.

Тобто бувають два види констант:

- 1) поіменовані, і
- 2) не поіменовані – літерали.

При створенні константи:

- 1) під неї виділяється ділянка оперативної пам'яті необхідного розміру,
- 2) до неї обов'язково заноситься значення цієї константи, і
- 3) у випадку поіменованої константи – йому присвоюється ім'я.

У цьому сенсі константа дуже схожа на змінну. Різниця полягає лише в тому, що в процесі виконання програми значення константи не змінюється, тобто що було записано у виділену під константу ділянку оперативної пам'яті при її створенні, то там і залишається до кінця виконання програми.

Тип даних визначає:

- 1) безліч значень, що ці дані можуть приймати, і
- 2) операції, що над ними можуть виконуватися.

У мові програмування VBA маються наступні типи даних: **String**, **Byte**, **Integer**, **Long**, **Currency**, **Single**, **Double**, **Date**, **Boolean** і **Variant**.

Для виведення різної інформації користувачеві в стандартному вікні повідомлення служить функція **MsgBox**. Вона є *стандартною (вбудованою)* функцією мови VBA. Інструкція процедури, в якій зустрічається ім'я функції **MsgBox**, здійснює її виклик чи звернення до неї, з «проханням» здійснити виведення повідомлення.

Однак таке «прохання» функція **MsgBox()** безпосередньо виконати не може – вона повинна знати, що саме потрібно вивести. Для цього при звертанні до неї передається, наприклад, рядок даних в подвійних лапках – "Здрастуй, Світ!". Таку додаткову інформацію, передану у функцію при її виклику, називають *аргументами* (чи *параметрами*). Перший параметр функції **MsgBox()** називається **Prompt (Запрошення)** і являє собою текст, який необхідно вивести. А для того, щоб указати VBA, що цей рядок є даними, які необхідно обробити (у нашому конкретному випадку, вивести на екран), а не ключовим словом мови, чи іншим компонентом програм, його треба взяти в лапки.

Найбільш простий спосіб одержання інформації від користувача – за допомогою функції **InputBox()**.



Функція в VBA – це особливий вид процедури, що повертає деяке значення.

Функція ***InputDialog()*** повертає дані типу ***String*** і має наступний прототип:

```
InputDialog( <запрошення>[ , <заголовок> ] ) ,
```

де: запрошення – обов'язковий текстовий рядок, що вказує користувачеві, яка інформація від нього очікується,

заголовок – не обов'язковий текстовий рядок, в який записаний заголовок вікна введення.


Функція введення ***InputDialog()*** працює наступним чином:

1. Відображає вікно введення із:

- зазначеним запрошенням,
- заголовком, а також
- полем введення і
- двома кнопками – **OK** і **Cancel**.

2. Чекає від користувача введення інформації в поле редагування.

3. Завершується:

- а) з підтвердження введеної інформації – за натисканням кнопки **OK**, чи
- б) зі скасуванням введеної інформації – за натисканням:
 - кнопки **Cancel**,
 - клавіші **Esc**,
 - кнопки закриття вікна  **Закрить**.

Внутрішні елементи даних програми бувають двох видів, називаються вони відповідно:

- 1) *змінні*, – які змінюють своє значення протягом виконання програми, і
- 2) *константи*, – які не змінюють свого значення протягом виконання програми.

Приклад. Написати процедуру, що вводить значення двох змінних цілого типу, а виводить – їхню суму.

Текст програми, що відповідає умовам прикладу, наведений далі:

```
Option Explicit
Sub Сума()
    Dim a As Integer, b As Integer, c As Integer
    a = InputBox ("Введіть a")
    b = InputBox("Введіть b")
    c = a + b
    MsgBox a & " + " & b & " = " & c
End Sub
```

У ньому:

Рядок 1 – забороняє неявне створення змінних.

Рядок 2 – заголовок процедури **Сума**.

Рядок 3 – створюються три змінні типу **Integer** а, b і c для збереження вхідних даних і результату, відповідно.

Рядки 4 і 5 – введення вхідних даних у змінні а і b.

Рядок 6 – обчислення суми і присвоювання її змінній c.

Рядок 7 – виведення результату.

Рядок 8 – завершення процедури.

Оформлення

Розрахунково-графічне завдання оформляють на аркушах формату А4 (210×297мм) через півтора інтервали. Текст може бути написаний як від руки, так і надрукований на комп'ютері. Його обсяг не повинний перевищувати 10 сторінок. Варіант завдання відповідає номеру студента за списком групи. Усі варіанти завдань реалізують за допомогою лінійного обчислювального процесу. Звіт за розрахунково-графічним завданням повинно бути наведений відповідно до наступної структури:

1. Титульний аркуш.
2. Зміст.
3. Вступ.
4. Рішення.
5. Висновок.
6. Список літератури.

На титульному аркуші вказують назву роботи, варіант, прізвище і групу студента. У Вступі необхідно показати основні особливості мови програмування VBA, область застосування, її достоїнства і недоліки. У розділі Рішення наводять постановку задачі, її рішення у вигляді програми на мові VBA, опис тексту програми й особливості реалізації. У Висновку необхідно зробити висновки по виконаній роботі.

Варіанти завдань

Для усіх варіантів! *Вхідні дані ввести з клавіатури, а результат вивести на дисплей.*

1. **3 градусів у радіани.** Написати програму переведення величини кута з градусів у радіани. $2\pi \text{ рад} = 360^\circ$.

2. **З радіанів у градуси.** Написати програму переведення величини кута з радіанів у градуси. $360^0 = 2\pi$ рад.
3. **З дюймів у метричну систему.** Написати програму переведення довжини відрізка, зазначеної в дюймах, у метричну систему, тобто виразити її в метрах, сантиметрах і міліметрах (1 дюйм = 2.54 см).
4. **З метричної системи в дюйми.** Написати програму переведення довжини відрізка, зазначеної в метричній системі, тобто в метрах, сантиметрах і міліметрах, у дюйми (1 дюйм = 2.54 см).
5. **Часовий інтервал.** Написати програму, що обчислює довжину інтервалу часу, якщо його початок і закінчення задані в годинах, хвилинах і секундах (у межах однієї доби). Результат вивести в тих же одиницях виміру.
6. **Комерція.** Написати програму, що обчислить, через скількох років підприємець накопичить необхідну суму грошей Y , якщо наданий момент він має X грошових одиниць, а щомісячний приріст капіталу складає $Z\%$.
7. **Квадратне рівняння.** Написати програму, що обчислює корені квадратного рівняння з позитивним дискримінантом, заданого своїми коефіцієнтами. Визначити так само погрішність обчислення, підставивши отримані значення у вхідне рівняння.
8. **Російські неметричні одиниці довжини.** Написати програму перекладу довжини відрізка, задану в метрах, у російську не метричну систему одиниць довжини, у якій 1 верста = 500 сажень, 1 сажень = 3 аршини, 1 аршин = 16 вершків, а 1 вершок = 44.45 мм.
9. **Вершина параболи.** Написати програму, що обчислює координати вершини параболи $y = ax^2 + bx + c$.
10. **Наближення Sin x.** Функція $y = \text{Sin } x$ на відрізку $[0; \pi/2]$ добре апроксимується розкладанням: $y = x - x^3/6 + x^5/120$. Написати програму, яка для заданого значення аргументу x обчислює значення функції y за цією формулою і порівнює з точним значенням, обчисленим за допомогою стандартної функції $\text{Sin}()$.

11. **Таксі.** У таксі одночасно сіли три пасажирів. Коли вийшов перший пасажир, на лічильнику було p_1 грошових одиниць, коли вийшов другий – p_2 , а останній – p_3 ; плата за посадку складає p_0 грошових одиниць. Написати програму, яка порахує, скільки повинен заплатити кожний з пасажирів.
12. **Середнє арифметичне.** Написати програму, що обчислює середнє арифметичне чисел x , y і z .
13. **Площа трикутника.** Написати програму, що обчислює площу трикутника зі сторонами a , b і c .
14. **Обмін значеннями.** Поміняти місцями значення змінних x , y і z так, щоб у x виявилось значення змінної y , у y – значення змінної z , а в z – колишнє значення змінної x .
15. **Дробова частина числа.** Змінній d присвоїти дробову частину позитивного числа x .
16. **Третя цифра від кінця.** Присвоїти цілій змінній i третю від кінця цифру в записі цілого позитивного числа j . Наприклад, якщо $j = 657876$, то $i = 8$.
17. **Перша цифра з дробової частини.** Присвоїти цілій змінній i першу цифру з дробової частини позитивного числа з плаваючою крапкою x . Наприклад, якщо $x = 123.456$, то $i = 4$.
18. **Сума цифр тризначного числа.** Цілій змінній i присвоїти суму цифр цілого тризначного числа j . Наприклад, якщо $j = 567$, то $i = 18$.
19. **Радіус.** Написати програму, що обчислює довжину окружності, площу круга, а також площу й обсяг кулі того самого заданого радіуса.
20. **Прямокутний трикутник.** Написати програму, що обчислює периметр і площу прямокутного трикутника по довжинах двох катетів.
21. **Добуток цифр чотиризначного числа.** Знайти добуток цифр заданого чотиризначного числа.
22. **Зворотний порядок цифр числа.** Визначити число, отримане виписуванням у зворотному порядку цифр заданого тризначного числа.

23. **Годинна стрілка – кут.** Визначити кут f (у градусах) між положенням годинної стрілки на початку доби і її положенням у h годин, m хвилин і s секунд ($0 \leq h \leq 11, 0 \leq m \leq 59, 0 \leq s \leq 59$).
24. **Годинна стрілка – години й хвилини.** Визначити повну кількість годин h і повну кількість хвилин m , що пройшли від початку доби до поточного моменту (у першій половині дня), коли годинна стрілка повернулася на f градусів (f – дійсне, і $0 \leq f \leq 360$).
25. **3 фунтів у кілограми і грами.** Написати програму переведення ваги, заданої у фунтах, у кілограми і грами (1 фунт = 405.9 грама).
26. **3 кілограмів і грамів у фунти.** Написати програму перекладу ваги, заданої в кілограмах і грамах, у фунти (1 фунт = 409.5 грама).
27. **Покупка.** Написати програму, що обчислює вартість покупки, яка складається з декількох зошитів, обкладинок до них і олівців.
28. **Поїздка на дачу.** Написати програму, що за заданою відстанню до дачі, витратою бензину на 100 км. пробігом автомобіля і вартістю 1 літра бензину визначає вартість поїздки туди і назад.
29. **Грошовий формат.** Написати програму, що перетворить задане дробове число в грошовий формат. Наприклад, 12.5 повинно бути перетворене до виду 12 грн. 50 коп.
30. **Площа кільця.** Написати програму, що визначає площу кільця.
31. **Внесок.** Написати програму, що за заданою річною процентною ставкою і часом збереження в днях, обчислює суму доходу за внеском.
32. **3 гривні в долари.** Написати програму, що на основі поточного курсу переводить суму, зазначену в гривнях, у долари США.
33. **3 доларів у гривні.** Написати програму, що на основі поточного курсу переводить суму, зазначену в доларах США, у гривні.

34. **Обсяг текстового файлу.** Написати програму, що визначає обсяг файлу в Кбайт, якщо рукопис складається із зазначеної кількості сторінок тексту. При цьому на кожній сторінці розташована однакова кількість рядків, і кожен рядок, у свою чергу, складається з однакової кількості символів, а один символ відповідає 1 байту інформації.
35. **Космічний корабель.** Написати програму, що визначає, за скільки років космічний корабель долетить до деякої планети, якщо відстань до неї складає 4.3 світлових років при заданій (км/с) швидкості космічного корабля, якщо швидкість поширення світла дорівнює 300000 км/с.

Список літератури

1. Васильев А., Андреев А. VBA в Office 2000: учебный курс. – СПб.: Питер, 2001. – 432 с.: ил.
2. Гарнаев А.Ю. Excel, VBA, Internet в экономике и финансах. – СПб.: БХВ-Петербург, 2001. – 816 с.: ил.
3. Гарнаев А.Ю. MS Excel 2002: разработка приложений. – СПб.: БХВ-Петербург, 2003. – 769 с.: ил.
4. Гарнаев А.Ю. Использование MS Excel и VBA в экономике и финансах. – СПб.: БХВ-Санкт-Петербург, 1999. – 336 с.: ил.
5. Долженков В.А., Колесников Ю.В. Microsoft Excel 2000. – СПб.: БХВ-Петербург, 1999. – 1088 с.: ил.
6. М. Коттингхэм Excel 2000: руководство разработчика: Пер. с англ. – К.: Издательская группа BHV, 2000. – 704 с.
7. Матросов А.В. и др. MS Office XP: разработка приложений / Матросов А.В., Новиков Ф.А., Усаров Г.Е., Харитонов И.А. / Под ред. Ф.А. Новикова. – СПб.: БХВ-Петербург, 2003. – 944 с.: ил.
8. Саймон, Джинжер Программирование в Excel: наглядный курс создания интерактивных электронных таблиц. : Пер. с англ. – М. : Издательский дом «Вильямс», 2002. – 336 с.: ил.
9. Самоучитель программирования на VBA в Microsoft Office / Хорев В.Д. – К.: Юниор, 2001. – 296 с., ил.
10. Харрис, Метью Освой самостоятельно программирование для Microsoft Excel 2000 за 21 день.: Пер. с англ. : Уч. пособие – М.: Издательский дом “Вильямс”, 2000. – 880 с.: ил.

Зміст

	Стр.
Вступ.....	1
Основні теоретичні положення.....	4
Оформлення	14
Варіанти завдань	14
Список літератури	19

НАВЧАЛЬНЕ ВИДАННЯ

Методичні вказівки для виконання розрахунково-графічної роботи з курсу "Економічна інформатика" (для студентів 1 курсу денної форми навчання бакалаврів напрямку 6.030504 "Економіка підприємства" і 6.030509 "Облік і аудит").

Укладач: Борис Іванович Погребняк

Редактор: М. З. Аляб'єв

Верстка: І.В. Волосожарова

План 2009, поз. 535М

Підп. до друку 27.04.2009	Формат 60x84 1/16.	Папір офісний.
Друк на різнографі	Умовн.-друк. арк. 0,9	Обл.- вид. арк. 1,2
Тираж 50 прим.	Замов. №	.

61002, Харків, ХНАМГ, вул. Революції, 12,

Сектор оперативної поліграфії ЦНІТ ХНАМГ

61002, Харків, вул. Революції, 12