

К.Ю. Островська, Є.В. Островський

РОЗРОБКА МОДУЛЮ РОЗШИРЕННЯ КЕРУВАННЯ МАКРОСАМИ ДЛЯ MICROSOFT EXCEL ЗАСОБАМИ C#

Анотація. Розроблено модуль розширення керування макросами для Microsoft Excel засобами C#, який дозволяє запускати макроси з використанням вхідних параметрів та керувати переліком макросів які знаходяться в документі.

Ключові слова: розробка, формат, Excel-DNA, мова, засоби, макрос, модуль, параметри, Microsoft Excel, функції користувача.

Microsoft Excel - це набагато більше, ніж просто електронна таблиця. З моменту впровадження в Excel 97 редактора Visual Basic і підвищення стабільності його роботи в Excel 2000. Ця програма стала надійною системою розробки додатків електронних таблиць. Програми, написані з використанням Excel, поряд з додатками Visual Basic, C++, Java, .NET і т.д., включаються в базовий набір засобів, що застосовуються для поліпшення функціональних можливостей основних програмних продуктів [1].

Розробка модулю

Для початку необхідно завантажити необхідні нам бібліотеки:

1. Excel-DNA.

<https://exceldna.codeplex.com/releases/view/616591>

Excel-DNA - це незалежний проект для інтеграції .NET рішень в MS Excel [1, 2].

Створення проекту. Запускаємо Visual Studio 2012 і створюємо проект, рисунок 1.

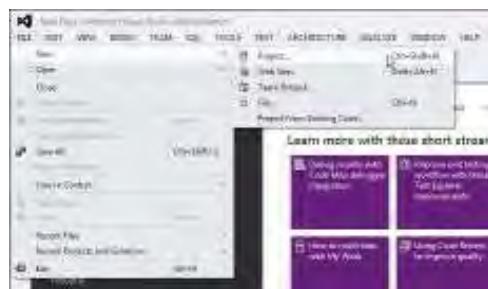


Рисунок 1 – Створити проект

4 (111) 2017 «Системные технологии»

У вікні майстра, вибираємо шаблон Templates / Windows. І тип проекту Class Library. Задаємо ім'я нашого проекту MacrosSelector, рисунок 2.

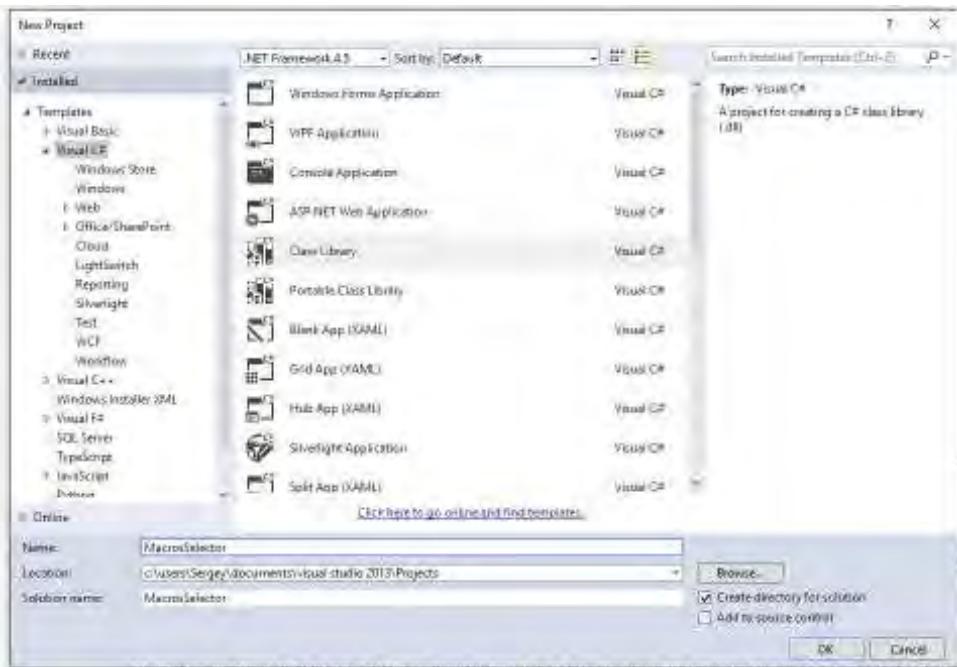


Рисунок 2 – Задати ім'я проекту

Коли проект створений, необхідно підключити потрібні нам бібліотеки, які ми попередньо завантажили. Для цього у вікні Solution Explorer на папці References натискаємо правою кнопкою миші і вибираємо пункт AddReference, рисунок 3. У вікні вибираємо вкладку Browse і додаємо бібліотеки, рисунок 4.

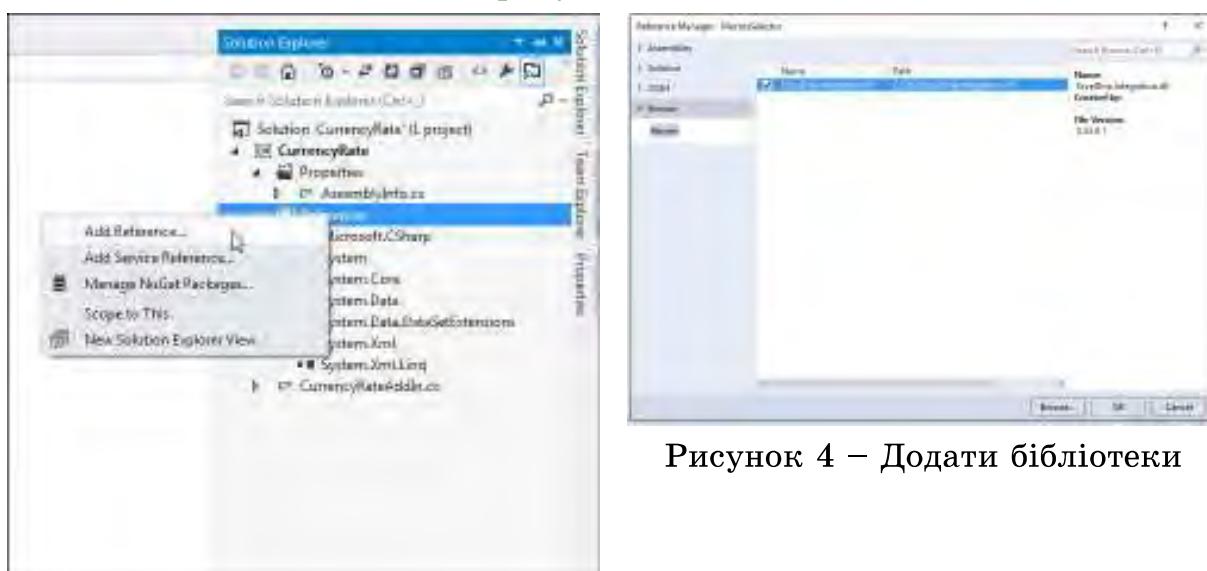


Рисунок 4 – Додати бібліотеки

Рисунок 3

Далі необхідно написання коду, на рисунку 5 представлено діаграму класів.

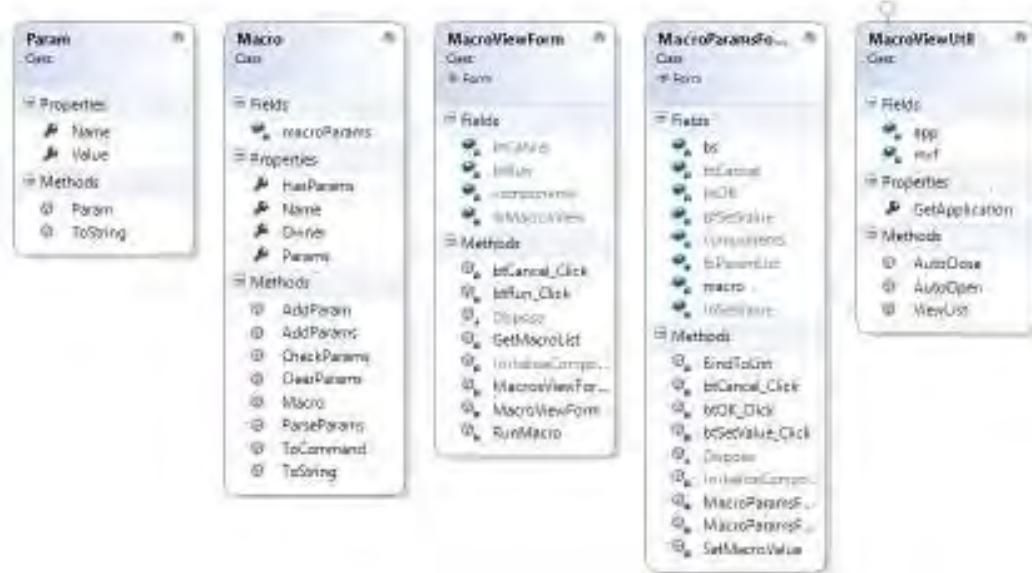


Рисунок 5 – Діаграма класів

Клас MacroViewUtil - це наш основний клас, його також можна назвати точкою входу в надбудову. В ньому реалізовані запуск плагіна, також кнопки для виклику нашого плагіна. Оскільки клас MacroViewUtil реалізує інтерфейс IExcelAddIn, ми в ньому повинні реалізувати методи AutoOpen() викликається при старті плагіна і AutoClose() викликається при видаленні плагіна. На рисунку 6 представлена реалізації цих методів.

```

public class MacroViewUtil : IExcelAddIn
{
    private static Excel.Application app = null;
    private static MacroViewForm mif = null;

    public void AutoOpen()
    {
        try
        {
            app = (Excel.Application)this.getOleObject();
            mif = new MacroViewForm();
        }
        catch (Exception ex)
        {
            System.Windows.Forms.MessageBox.Show(ex.Message, "Error", System.Windows.Forms.MessageBoxButtons.OK, System.Windows.Forms.MessageBoxIcon.Error);
        }
    }

    public void AutoClose()
    {
        app = null;
    }
}

```

Рисунок 6 – Реалізація методів

Для розміщення на панелі Ribbon кнопки, ми повинні вказати в методі атрибут ExcelCommand і вказати параметри для цього атрибута, наприклад ім'я нашої кнопки, поєднання клавіш для швидкого виклику і т.д., рисунок 7.

```
[ExcelCommand(MenuName = "Macros Viewer", MenuText = "Show macros list", ShortCut = "^m")]
public static void ViewList()
{
    if (app == null)
    {
        System.Windows.Forms.MessageBox.Show("Instans of Excel application is null!", "Error", System.Windows.Forms.MessageBoxButtons.OK, System.Windows.Forms.MessageBoxIcon.Error);
        return;
    }
    mvf.ShowDialog();
}
```

Рисунок 7

Клас MacroViewForm - це клас вікна, який буде відображати список макросів, які містяться в поточній робочій книзі Excel.

MacroParamForm - це клас вікна, який буде відображати список параметрів для макросу з можливістю задати значення для обраного параметра.

Клас Macro - власне сутність, яка представляє макрос.

Param - це клас, який описує параметр для макросу.

Коли все готово, нам необхідно створити файл CurrencyRate.dna. Це файл в xml форматі, де ми вказуємо, яку саме зовнішню бібліотеку необхідно завантажити. Вміст файлу представлено на рисунку 8.

```
1 <DnaLibrary Name="Macros Selector" RuntimeVersion="v4.0">
2     <ExternalLibrary Path="MacrosSelector.dll" />
3 </DnaLibrary>
```

Рисунок 8 - Вміст файлу CurrencyRate.dna

Також потрібно додати файл *.xll, який поставляється разом з бібліотекою ExcelDNA.Integration.dll.

Далі налаштуємо наш проект на налагодження. Для цього перейдемо в властивості проекту та відкриємо вкладку Debug. Необхідно вказати, що будемо використовувати зовнішню програму і в поле введення вкажемо шлях до Excel.exe. В поле Command line arguments необхідно вказати шлях до нашого MSelectorAddIn.xll (якщо у нас використовується офіс x64 то до імені потрібно додати значення розрядності MSelectorAddIn64.xll). На рисунку 9 показана вкладка Debug вікна властивостей проекту.

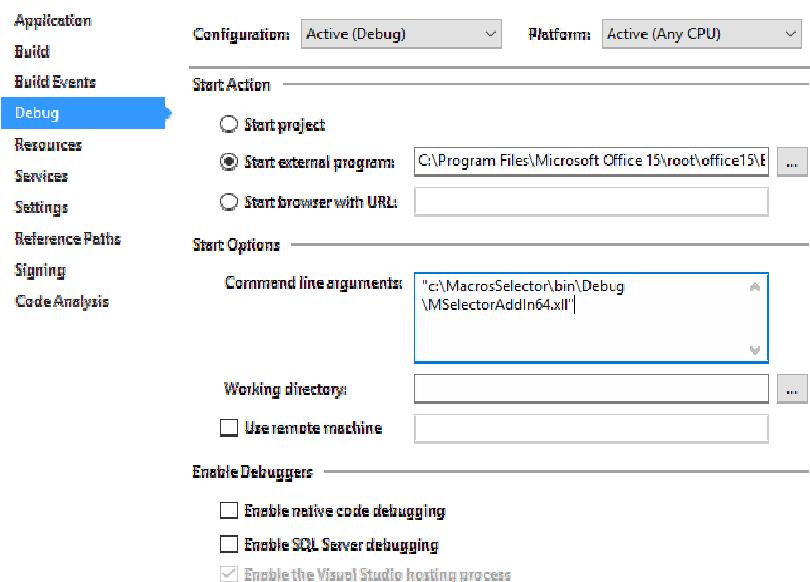


Рисунок 9 - Вкладка Debug вікна властивостей проекту

В результаті ми можемо запустити наш проект на налагодження і перевірити роботу розширення для Excel. Але для початку потрібно налаштувати Excel для запуску макросів.

У вікні, в списку зліва вибираємо «Центр управління безпекою». А в правій частині натискаємо на кнопку «Параметри центру управління безпекою».

VBA вважається стандартною мовою написання сценаріїв для додатків Microsoft, і в даний час він входить до складу всіх додатків Office і навіть додатків інших компаній. Отже, оволодівши VBA для Excel, ви зможете відразу перейти до створення макросів для інших програмних продуктів Microsoft. Більш того, ви зможете створювати повноцінний програмний продукт, одночасно використовуючи функції самих різних додатків.

Включити макроси в Excel.

За замовчуванням, вкладка, що відповідає за управління та наявністю макросів в Excel, прихована. Щоб активувати цю опцію перейдіть по вкладці «Файл» в групу «Параметри». У діалоговому вікні Параметри Excel, перейдіть по вкладки «Налаштування стрічки», в правому полі зі списком ставимо маркер напроти вкладки «Розробник», рисунок 10. Дані дії актуальні для версій Excel 2010 і старше.

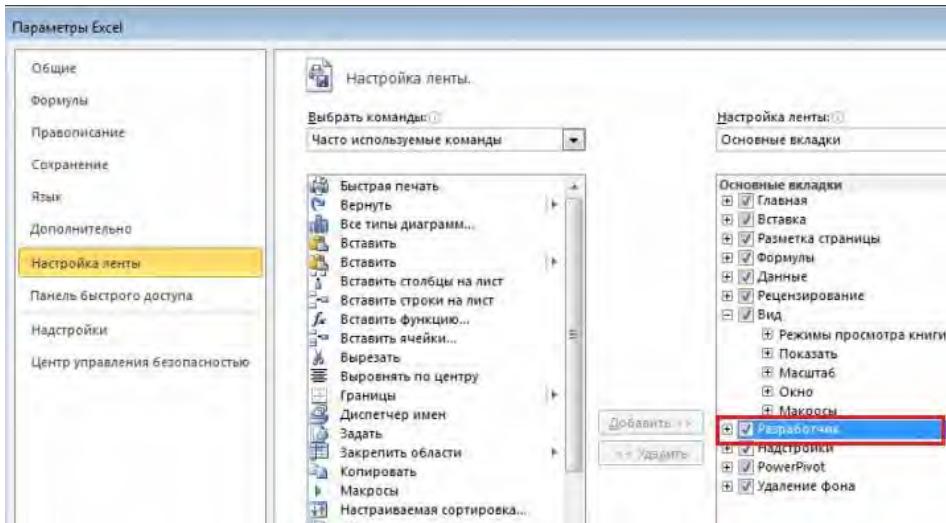


Рисунок 10 - Відображення вкладки «Розробник»

Написання макросів в Excel.

У вкладці «Розробник» у групі «Код», натисніть кнопку «Запис макросу». З'явиться діалогове вікно «Запис макроса», яка запи- тує деяку інформацію про майбутнє записуваному коді. З даного мо- менту Excel буде фіксувати кожну дію користувача в модулі VBA, будь то введення даних, форматування або створення діаграм. Щоб зупинити запис макросу, натисніть кнопку «Зупинити запис».

Тепер є можливість подивитися список всіх створених макро- сів, натиснувши на кнопку «Макрос», що знаходиться в групі «Код». У діалоговому вікні ви можете дати більш описові імена своїм кодами або задати поєднання клавіш, які б запускали той чи інший макрос.

Редагування макросів. Необхідно перевірити, який код згене- рував нам Excel. Згенерований код написаний на мові VBA (Visual Basic for Applications). Щоб побачити її потрібно відкрити Редактор VB (VBE). Пишемо макроси, рисунок 11.

```

Sub SayHello()
    MsgBox ("Hello!")
End Sub

Sub SayBy(sMs As String, num As Integer)
    MsgBox ("See next time " + sMs + "!")
End Sub

```

```

Sub SayBy4(name As String)
    MsgBox ("See next time " + name + "!")
End Sub

```

Рисунок 11 – Написання макросу

Нижче представлені два макроси:

- макрос SayHello - який не приймає параметрів і видає вікно повідомлення з привітанням;
- макрос SayBy4 - який приймає, як параметр строкову змінну і також виводить вікно повідомлення з текстом прощання в якому буде виведено ім'я, яке ми вкажемо в параметрі [3].

Отже запускаємо наш проект на налагодження. Переходимо на вкладку «Надбудови» і бачимо нашу кнопку «Macros Viewer», натискаємо, і в випадаючому меню з'являється команда «Show macros list», рисунок 12.

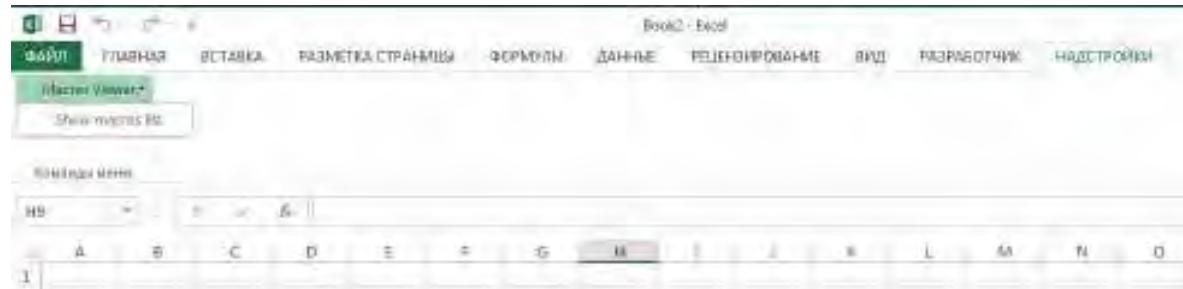


Рисунок 12

Після натискання на команду з'являється вікно зі списком макросів в активній робочій книзі, рисунок 13.

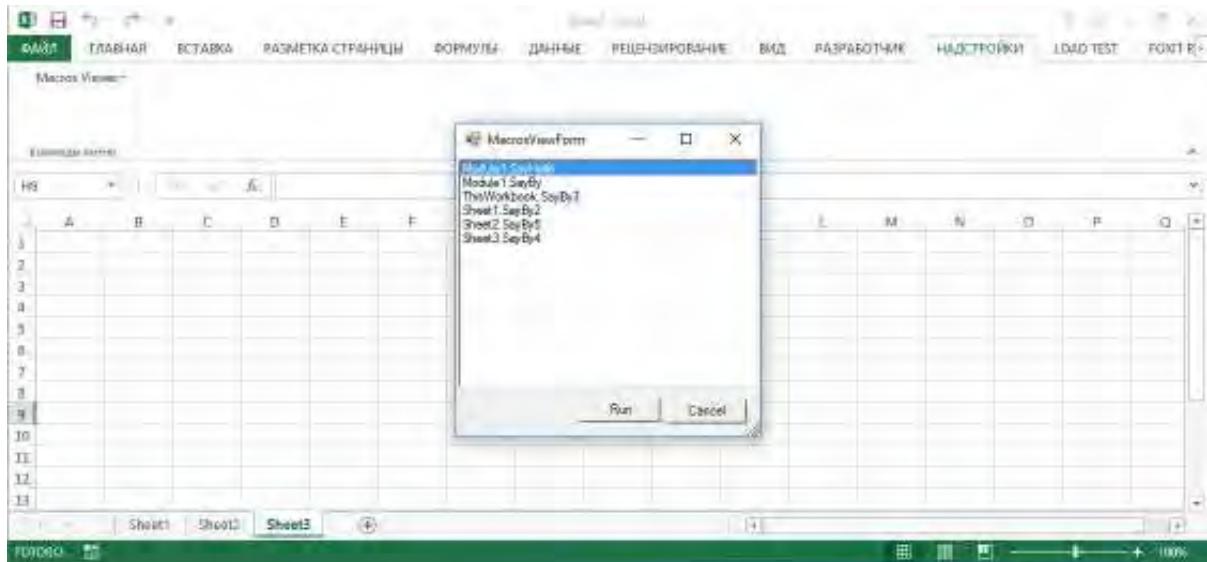


Рисунок 13

Для запуску макросу досить виділити макрос у списку, і натиснути кнопку «Run». Нижче на рисунку 14 показаний запуск макросу SayHello.



Рисунок 14 – Запуск макросу SayHello

Далі, запустимо макрос з параметром з'явиться вікно зі списком параметрів.

Для того, щоб задати параметр, треба виділити параметр в списку ввести значення і натиснути кнопку Set, рисунок 15.

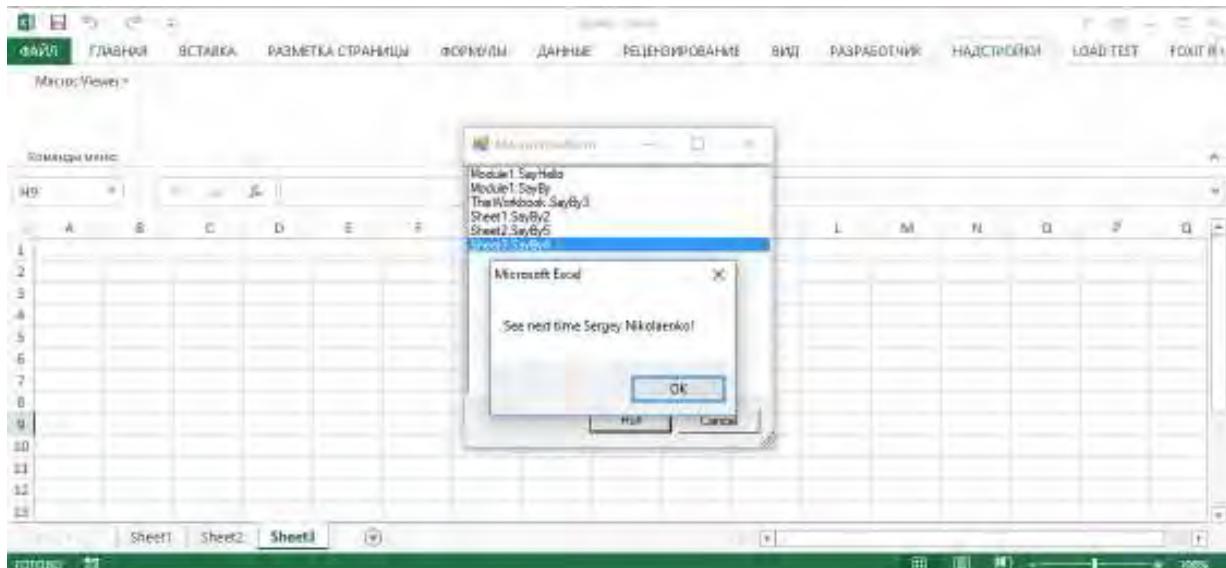


Рисунок 15

Висновки

В результаті розроблено модуль розширення керування макросами для Microsoft Excel засобами С #, який надає можливість запуску макросів з вхідними параметрами, бо в стандартному інструменті такої можливості немає.

ЛІТЕРАТУРА

1. Джон Уокенбах Excel 2013: Профессиональное программирование на VBA. - М.: «Диалектика», 2014. — 960 с.
2. Посилання в мережі Інтернет: <http://json.org/json-ru.html>
3. Стивен Буллен, Роб Боуви, Джон Грин Excel и VBA: Профессиональная разработка приложений. – Вильям, 2007.- 736 с.