

О ВЫБОРЕ БАЗОВЫХ WINDOWS — ПРИЛОЖЕНИЙ ДЛЯ ОБУЧЕНИЯ ВЫЧИСЛЕНИЯМ И ПРОГРАММИРОВАНИЮ

В. Г. Афонин

Введение. Высокоэффективное использование современной компьютерной техники является важнейшим компонентом подготовки студентов технических, экономических, естественнонаучных и многих других специальностей. Особую роль при этом играет вычислительная и программистская подготовка квалифицированных пользователей. В идеале большинство студентов должно проходить обучение вычислениям и программированию на базе одних и тех же программных средств. В этом случае можно разработать высококачественное унифицированное методическое обеспечение и добиться высокого уровня подготовки специалистов.

Автор предлагает в качестве таких «типовых» программных средств систему компьютерной математики (СКМ) MathCAD и табличный процессор Excel со встроенной (во все приложения MS Office) средой программирования Visual Basic for Applications (VBA).

1. Обучение вычислениям.

Решение задач вычислительного характера должно играть центральную роль в компьютерной и математической подготовке студентов многих специальностей. Студент должен научиться на высоком уровне проводить как простые вычисления (по обычным формулам), так и достаточно сложные расчеты. При этом каждый студент обязан приобрести знания, умения, навыки и привычки проверять все числовые и символьные результаты [1].

Для этого студент должен в достаточной степени овладеть по крайней мере двумя вычислительными программными средствами для символьных и численных вычислений.

По мнению автора, такими «стандартными» средствами должны быть MathCAD и Excel + VBA. Ниже приводится обоснование такого выбора.

1.1. Об использовании Excel + VBA в вычислительной компьютерной подготовке.

Основным и весьма распространенным приложением MS Office для выполнения вычислений и графического отображения данных является Excel. С ним студенты знакомятся еще в школе, но, как правило, совершенно недостаточно владеют вычислениями даже по сравнительно простым формулам. И практически все выпускники школ незнакомы со средой программирования VBA, которая многократно увеличивает возможности Excel при решении сколько-нибудь сложных задач.

Как Excel, так и язык VB имеют дело со «строчной» формой записи числовых выражений. И здесь недостаточно подготовленный пользователь с немалой вероятностью может совершить грубейшую ошибку даже в записи простейших выражений. Об этом говорит многолетний опыт преподавания автора и его коллег в вузе.

Например, при вычислении по формуле $= -2^2$ в ячейке Excel мы получим в результате 4, тогда как во всех других известных автору вычислительных средах результат будет равен -4 . Разумеется, приведенный пример тривиален, и ошибка сразу видна. Но в более сложной ситуации, когда выражение вида $-x^2$, стоящее вначале более сложной формулы, получает положительное значение, такая необычная трактовка этого выражения в Excel вполне может быть не замечена со всеми вытекающими последствиями.

Отсюда следует элементарный вывод: вычисления даже по несложным формулам весьма желательно (а скорее даже необходимо!) проверять. Эта проверка может осуществляться разными способами. «Простейший» из них — с помощью микрокалькулятора — устраняет далеко не всех и далеко не всегда. Проверка с использованием других вычислительных сред — более эффективная процедура. Например, вычисляя выражение —

x^2 с помощью VBA, мы вышеуказанную ошибку обнаружим сразу. Однако, поскольку, и в VBA и в формулах рабочего листа Excel запись формул осуществляется в строку, многие ошибки с помощью такой проверки не будут обнаружены. Например, не будет выявлена весьма распространенная ошибка, когда, вместо записи вида $A/(B*C)$ осуществляется запись $A/B*C$, которая эквивалентна записи $(A/B)*C$. Такого рода ошибки легко видеть при работе в системах, где формулы, содержащие числители и знаменатели, представляются в общематематической форме записи, в виде «этажных» дробей. Это прежде всего СКМ типа MathCAD, Derive и другие. (Кстати, весьма простую и компактную систему Derive очень удобно использовать для работы со «строчными» формулами. Там легко получить как «строчную», так и «этажную» форму записи любого, даже весьма сложно-го выражения.)

Отметим теперь следующие положительные моменты при использовании Excel совместно с VBA.

В Excel имеется весьма обширный набор математических, статистических и других функций рабочего листа, тогда как в VB такой набор следует признать недостаточным.

Но многие из встроенных функций рабочего листа можно либо использовать для проверки вычислений в VBA, либо попросту включать в программный код VBA. Например, с приставкой Application или WorksheetFunction в числовые выражения VBA можно включать отсутствующие в VB гиперболические и обратные тригонометрические функции, а также довольно сложные статистические. Так, в VBA допустима запись WorksheetFunction.GammaDist(<список аргументов>) — интегральная функция Гамма-распределения или Application.Log(x, a) - логарифм по любому основанию или Application.Acos(x) — арккосинус.

Более того, в программный код VBA фактически можно включать мощную вычислительную надстройку «Поиск решения» (Solver), позволяющую решать задачи математического программирования, системы нелинейных уравнений и другие достаточно сложные задачи.

Далее, при работе с Excel + VBA легко решаются проблемы графического отображения числовых данных, форматирования (в том числе и условного), ввода - вывода данных на рабочий лист. Эта последняя проблема весьма просто решается на основе использования объектов Range (для работы в режиме адресации типа A1) и Cells (для работы в режиме адресации типа R1C1).

В то же время, вычислительные процедуры, сформированные в VBA, весьма существенно повышают возможности электронных таблиц.

1.2. Об использовании СКМ MathCAD в вычислительной компьютерной подготовке.

MathCAD является достаточно простой, весьма наглядной и, вероятно, самой популярной вычислительной средой общего назначения. MathCAD можно рекомендовать к изучению даже старшеклассникам. Инженеры и экономисты, представители других специальностей могут эффективно используют MathCAD для решения самых разнообразных профессиональных задач. Среди положительных сторон следует отметить и возможность разработки и эффективного использования библиотеки пользовательских функций. При этом в качестве аргументов можно использовать и имена других функций.

Но и здесь имеется немало нюансов, знание которых избавляет пользователя от получения неверных результатов. Например, при использовании вычислительной процедуры Given/Find для решения вырожденной системы линейных алгебраических уравнений, имеющей бесконечное множество решений, выдается только одно из этих решений. Тем самым, недостаточно квалифицированный пользователь фактически вводится в заблуждение. Есть и множество других деталей, значительная часть которых отражена в опубликованных источниках.

Изучение MathCAD в вузе должно начинаться как можно раньше. В идеале MathCAD должен полностью сопровождать курс высшей математики, а также курсы физики, механики, и других дисциплин. Это позволит существенно расширить круг задач, решаемых в этих дисциплинах и поднять уровень подготовки специалистов. Аналогичное сопровождение можно осуществлять и в школе.

2. Обучение программированию.

В школах и даже в вузах обучение «реальному» программированию ведется чаще всего обычных средах программирования на языках семейства Pascal и Basic, реже — C и Fortran.

Практика, однако, показывает, что в дальнейшем подавляющее большинство пользователей не работает в этих средах программирования. Это относится в равной мере и к Qbasic, и к VB6, и к Turbo Pascal, и к Delphi... Таким образом, студентами теряется значительная часть полученных знаний, умений и навыков. В то же время, необходимость обучения элементам алгоритмизации и программирования на территории СНГ можно считать общепризнанной. Такое обучение целесообразно как для общего развития, так и для высокоэффективного использования современных программных продуктов.

Известно, что Excel является вторым (после Word) по популярности программным продуктом. Поэтому вероятность того, что пользователю придется работать с Excel, достаточно велика.

Известно также, что в современных руководствах по высокоэффективному использованию Excel значительное внимание уделяется программированию в VBA, так как в комплексе с VBA Excel становится весьма мощным средством для решения самых разнообразных задач, в том числе и для создания Windows — приложений.

В то же время, при использовании Excel на предприятиях и организациях пока еще сравнительно редко используют VBA.

Поэтому пользователи, освоившие в должной мере работу с Excel и VBA, помимо всего прочего, смогут существенно повысить свою конкурентоспособность на рынке труда.

Возможности и особенности среды программирования VBA весьма близки к VB6, за исключением, быть может, графики, которая, однако, в достаточной степени присутствует в Excel и при необходимости, может быть задействована с использованием макросов.

При изучении VBA на базе стандартных модулей можно минимизировать усилия пользователей, направленные на изучение форм, элементов управления, непростых процедур ввода-вывода и других составляющих процесса разработки приложений, весьма характерных для работы в среде типа VB6. Высвободившееся учебное время можно с достаточно большим эффектом использовать для получения практически важных навыков работы в Excel, для усиления алгоритмической и математической подготовки, для работы с важнейшими структурами языка программирования.

Таким образом, решение и оформление задач вычислительно - графического характера в Excel с VBA может осуществляться значительно проще, чем через создание в VB6 — Windows - приложений.

Знакомство с элементами программирования и, в частности, с ветвлениями и циклами типа **For** и **While** вполне можно начинать и в MathCAD. Наличие ранжированных переменных, массивов, компактность и высокая наглядность программных блоков (ПБ) позволяет сравнительно просто познакомить пользователя с циклическими вычислительными процессами. Кроме того, использование ПБ существенно расширяет вычислительные возможности MathCAD. Однако имеются и недостатки. Так, в ПБ нельзя включить вычислительный блок **Given**, не предусмотрены комментарии. В ПБ, как и вообще в MathCAD, фактически нет типов данных: можно сказать, что вычисления ведутся только с переменными типа Variant. Вообще, программирование в MathCAD заметно отличается от традиционного. Отсутствуют и эффективные средства отладки, нет достаточно хорошей диагностики ошибок.

Этих недостатков лишена среда VBA. Как уже говорилось выше, VBA+Excel допускает использование такой мощной вычислительной процедуры, как Solver. Кроме того, вычисления в VBA могут проводиться значительно быстрее, чем в MathCAD (например, за счет использования переменных типа Single). Благодаря только этим особенностям, VBA+Excel может оказаться предпочтительнее MathCAD при решении целого ряда задач.

Освоив основы работы в VBA, можно легко и просто перейти к другим средам программирования.

Таким образом, если в начале курса компьютерной подготовки дается MathCAD, то обучение программированию можно вести по схеме: ПБ MathCAD, VBA+Excel, а затем VB6. А для студентов, проявивших способности к программированию и использованию ЭВМ для вычислений, можно читать факультативные курсы по изучению других программных продуктов.

В заключение еще раз отметим, что, выбрав типовые программные средства и разработав на их основе высококачественное методическое обеспечение для различных учебных дисциплин, как в средних, так и высших учебных заведениях, можно добиться существенного повышения качества подготовки квалифицированных специалистов [2].

Список источников

1. *Афонин, В.Г.* О математической и компьютерной подготовке студентов [Текст]/ В. Г. Афонин // Вестник Брестского государственного технического университета. Физика, математика, информатика. — 2004, №5 (29) / Брест, БГТУ, 2005. - С.13—15.
2. *Афонин, В.Г.* О путях создания и совершенствования методического обеспечения учебного процесса [Текст] / В. Г. Афонин // Материалы Международного форума «ОБРАЗОВАНИЕ ДЛЯ УСТОЙЧИВОГО РАЗВИТИЯ: НА ПУТИ К ОБЩЕСТВУ ЗНАНИЯ», Минск, 5—6 апреля 2005 года, стр. 492—495. Материалы Международного форума «Образование для устойчивого развития: на пути к обществу знания», Минск, 5—6 апреля 2005 года / Минск, 2005. — С. 492—495