

# Преподавание офисного программирования с использованием технологии открытого образования

*Офисное программирование предназначено для повышения производительности труда конечного пользователя. Коммерческие прикладные программные продукты включают инструментарий, предназначенный для пользовательского программирования. Однако, в настоящее время, в России, данные возможности повышения производительности труда не используются в должной мере.*

**Ключевые слова:** *Офисное программирование, языки пользовательского программирования, встроенные языки прикладных экономических систем, повышение производительности труда, открытое образование.*

## TEACHING OFFICE PROGRAM BY USING TECHNOLOGIES FOR OPEN EDUCATION

*Office programming is designed to enhance productivity of the end user. Commercial application software products include tools designed for custom programming. However, at the present time, in Russia, the possibility of improving productivity is not utilized to the extent.*

**Keywords:** *Office programming, custom programming languages, embedded languages for applied economic systems, increase productivity, open education.*

### 1. Назначение офисного программирования

При переходе в информационное общество, т.е., общество в котором большинство работающих занято производством, хранением и переработкой информации, особое значение имеет повышение производительности труда специалистов, которые обеспечивают обработку экономической информации.

Одним из недостаточно используемых резервов повышения производительности труда у пользователей прикладных программных продуктов является офисное программирование.

Важным аспектом является то, что повышение производительности труда, как правило, необходимо обеспечивать у работающих сотрудников офисов. Таким образом, открываются широкие возможности использования открытого образования при обучению офисному программированию.

Офисное программирование, как правило, предназначено для автоматизации офисной деятельности. Основой офисного программирования является применение языков пользовательского программирования.

Языки пользовательского программирования (End-User Programming, EUP) предназначены для повышения производительности труда конечного пользователя. Иногда, вместо термина «языки пользовательского программирования», используется термин «языки пользовательского развития» (End-User Development, EUD).

Офисное программирование имеет ряд особенностей, которые отличают его от разработки приложений на универсальных языках программирования.

Прежде всего, офисное программирование предназначено для повышения производительности труда конечного пользователя при использовании прикладного программного продукта. Прикладное программное обеспечение может

быть использовано на двух основных уровнях:

Во-первых, на основе применения средств интерактивной обработки;

Во-вторых, с применением языка пользовательского программирования.

Применение прикладного программного обеспечения на уровне интерактивной обработки, как правило, позволяет выполнять основные функции прикладного программного продукта. Однако, эффективность применения прикладного программного обеспечения на данном уровне, может оказаться не достаточно высокой.

Применение языков пользовательского программирования позволяет достичь более высокий уровень производительности труда офисных работников.

В соответствии с технологией маркетинга, разработчики прикладных программных продуктов, предусматривают возможность использования языков пользова-



**Анжелика Витальевна Рычкова,**  
к.п.н., доцент, доцент кафедры  
Автоматизированных систем  
обработки информации и управления  
(АСОИиУ)  
Тел.: (495) 442-61-11  
Эл. почта: Rytchkova.AV@rea.ru  
ФГБОУ ВО «РЭУ им. Г.В. Плеханова»  
<http://www.rea.ru/>

**Anzhelika V. Rychkova,**  
Candidate of Pedagogical Sciences,  
Associate Professor, Associate Professor,  
Department of automated systems of  
information processing and management  
Tel.: (495) 442-61-11  
E-mail: Rytchkova.AV@rea.ru  
Russian University of Economics  
named after G. V. Plekhanov  
<http://www.rea.ru/>

тельского программирования. Характеристики средств, предусматривающих использование языка пользовательского программирования, могут в значительной степени отличаться в различных прикладных системах. Применение языка пользовательского программирования прикладной системы требует от пользователя знание основ информационных технологий.

Важной особенностью офисного программирования является невозможность формирования независимых приложений. Модули языка пользовательского программирования всегда описывают обработку объектов своей прикладной системы. Как правило, программы, написанные на языке пользовательского программирования, сохраняются в файловом формате того прикладного приложения, для которого написана программа [1, 2]. В частности, макросы VBA, созданные в Excel, сохраняются в файле рабочей книги Excel, а программы, написанные в текстовом редакторе Word, сохраняются в файле документа. Поэтому, для выполнения программы VBA требуется предварительно запустить приложение, в котором создана программа.

Наиболее широкое распространение среди пользователей получили языки пользовательского программирования программных продуктов Microsoft Office.

Программные продукты Microsoft Office имеют свой язык пользовательского программирования, который называется VBA (Visual Basic for Applications). VBA представляет собой язык программирования, предназначенный для расширения возможностей прикладных программных продуктов Microsoft Office, в частности MS Excel, MS Word, MS PowerPoint, MS Access, MS Outlook.

Язык VBA основан на концепции объектно-ориентированного программирования. Прикладное программное приложение Microsoft Office состоит из совокупности взаимосвязанных объектов. Характер взаимосвязей объектов, фиксируется архитектурой прикладного программного приложения. Архитектура программного приложения, прежде всего, зависит от операци-

онной системы, в которой разрабатывалось прикладное приложение.

Каждое из приложений Microsoft Office использует собственную систему объектов. Например, в текстовом редакторе Word используется объект Document, в электронной таблице Excel используются объекты Workbook и Worksheet. Следовательно, профессиональное программирование на языке VBA, предполагает умение правильно использовать объекты прикладных приложений, их свойства и методы.

Однако важной особенностью языка VBA является возможность эффективного программирования пользователями с различным уровнем знания технологий программирования. В простейшем случае, пользователь совершенно не знакомый с технологией объектно-ориентированного программирования, может создавать эффективно работающий программный код [3].

Кроме того, в VBA объектно-ориентированное программирование реализовано таким образом, что пользователь может использовать данную технологию обладая минимальными знаниями. В частности, очень удобно для пользователя организовано написание программных кодов разрабатываемых методов. Хорошо продуман отбор наиболее важных элементов на панель ToolBox, которая содержит элементы управления, помещаемые в экранную форму [4].

Принципиально важной является возможность развития используемого инструментария при офисном программировании. В частности, можно выделить следующие возможности.

Во-первых, возможность подключения дополнительных элементов управления на панель ToolBox. Прежде всего, элементы управления могут находиться в различных библиотеках, подключенных при установке системы. При необходимости могут быть дополнительно подключены требуемые библиотеки.

Во-вторых, имеется возможность, работая в одном из приложений, используя объекты любого приложения, управлять другим прикладным приложением Microsoft Office.



**Александр Алексеевич Смирнов**,  
к.э.н., доцент, доцент кафедры  
Автоматизированных систем  
обработки информации и управления  
(АСОИиУ)  
Тел.: (495) 442-61-11  
Эл. почта: Smirnov.AA@rea.ru  
ФГБОУ ВО «РЭУ им. Г.В. Плеханова»  
<http://www.rea.ru/>

**Alexander A. Smirnov**,  
Candidate of Economic Sciences,  
Associate Professor, Associate Professor,  
Department of automated systems of  
information processing and management  
Tel.: (495) 442-61-11  
E-mail: Smirnov.AA@rea.ru  
Russian University of Economics  
named after G. V. Plekhanov  
<http://www.rea.ru/>

В-третьих, объединить данные из нескольких приложений Microsoft Office в одном документе.

В-четвертых, возможность использования языка SQL (Structured Query Language, язык структурированных запросов).

В-пятых, существует возможность использования в офисных приложениях, внешних объектов, которые поддерживают технологии DDE, OLE, ActiveX и т.д.

## 2. Особенности формирования знания в процессе преподавания офисного программирования

Необходимо исходить из того, что формирование знаний представляет собой сложный и тяжелый процесс. Следует учитывать, что значительное число обучающихся офисному программированию должны составлять работающие сотрудники офисов. Следовательно, возрастает значение использования различных возможностей технологии открытого образования. Особое внимание, по-видимому, необходимо уделить возможности обращения за консультациями.

Формирование знаний в области языков пользовательского программирования целесообразно подразделить на отдельные сегменты.

В результате изучения каждого конкретного сегмента пользователь имеет возможность сформировать и практически использовать усвоенные знания. Разобравшись даже в одном, первом сегменте пользователь получает возможность, благодаря применению простейших конструкций повышать производительность своего труда. Таким образом, по мере приобретения навыков программирования, у пользователя постоянно расширяются возможности по более эффективно применению прикладной экономической системы.

Подразделение изучаемой информации по языкам пользовательского программирования на сегменты позволяет определить для каждого из сегментов наиболее важные акценты, которые позволяют пользователям лучше разобраться в особенностях компьютерной обработки экономической информации.

Кроме того, данный подход позволяет рассмотреть процедуру формирования знаний в динамике перехода от пользовательского программирования к программированию профессиональному.

В частности, изучение программирования на языке VBA целесообразно осуществлять в следующей последовательности сегментов (рисунок).

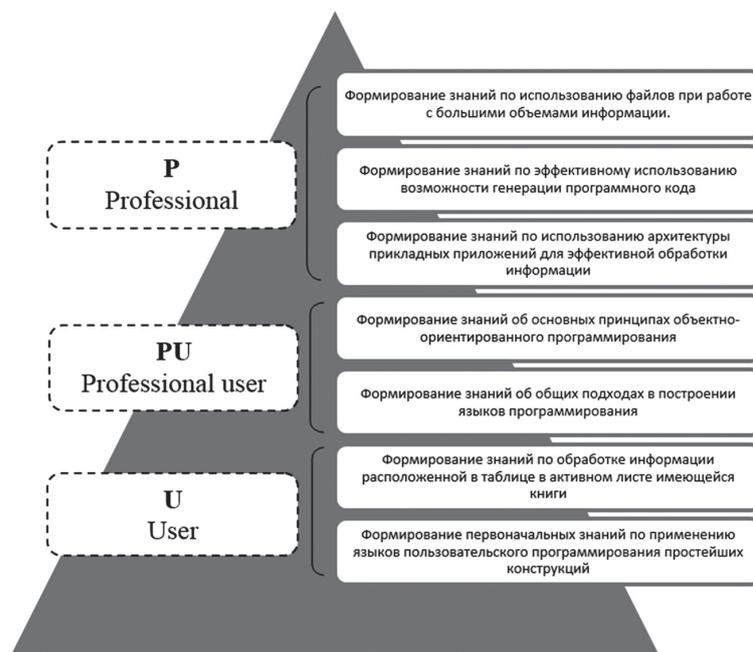


Рис. Последовательность формирования знаний пользователя при обучении офисному программированию

Сегмент первый. Программирование на VBA с использованием простейших конструкций.

Первый сегмент обучения обеспечивает формирование первоначальных знаний по применению языков пользовательского программирования. Данный сегмент ориентирован на пользователей Microsoft Office абсолютно не знакомых с программированием.

При формировании знаний первого сегмента требуется тщательный отбор необходимой информации. Следует учитывать, что первые шаги психологически очень трудные. Таким образом, при формировании знаний по первому сегменту требуется освоить небольшой объем информации, который тщательно отрабатывается на практических примерах.

По-видимому, при освоении первого сегмента, целесообразно остановиться на разборе канонических структур программирования. Из теории программирования известно, что знание всего трех канонических структур позволяет написать программу любой сложности [5].

При освоении материала первого сегмента, при использовании технологии открытого образования, могут потребоваться консультации по самым элементарным вопросам.

Сегмент второй. Программная обработка структурных типов.

Второй сегмент обучения обеспечивает формирование знаний по обработке информации расположенной в таблице в активном листе имеющейся книги. Освоение материала второго сегмента не требует больших способностей к программированию. Данный материал с успехом может освоить пользователь, который приобрел в программировании самые элементарные навыки. Задачи, разбираемые при изучении второго сегмента, имеют ярко выраженную экономическую направленность. При программировании задач данного сегмента необходимо делать акценты на проверке корректности представления исходной информации. При формировании знаний второго сегмента студенты экономических специальностей

должны понять важность и сложность корректного ввода информации.

При освоении материала второго сегмента, при использовании технологии открытого образования, значительную пользу могут принести открытые материалы экономических вузов.

Сегмент третий. Модульное программирование.

Изучение третьего сегмента ориентируется на формирование более глубоких знаний у тех пользователей, которые заинтересуются программированием. Изучение модульного программирования позволяет понять общие подходы в построении языков программирования, как встроенных в прикладные системы, так и универсальных языков программирования.

При освоении материала третьего сегмента, важно сопоставить материалы нескольких открытых ресурсов.

Сегмент четвертый. Программирование экранных форм.

Изучение четвертого сегмента связано с формированием знаний об основных принципах объектно-ориентированного программирования. Если учесть, что практически все, разрабатываемое в настоящий момент, прикладное программное обеспечение основано на объектно-ориентированных технологиях, то изучение этого сегмента будет очень полезно продвинутым пользователям. Следует учитывать, что изучение объектно-ориентированного программирования позволяет перейти к профессиональному программированию.

При освоении материала четвертого сегмента, необходимо использовать возможности консультаций. Визуальное программирование без посторонней помощи осваивается очень тяжело [6].

Сегмент пятый. Программирование с использованием архитектуры прикладного приложения.

Освоение пятого сегмента позволяет сформировать знания, которые открывают широчайшие возможности для программной обработки информации, расположенной в прикладных программных продуктах Microsoft Office. Ис-

пользование архитектуры прикладных приложений позволяет при обработке информации применять объекты, расположенные на различных уровнях. Однако материал пятого сегмента доступен, исключительно, тем пользователям, которые сформировали качественные знания по технологии объектно-ориентированного программирования и, таким образом, вышли на уровень профессионального программирования [7–9].

Характер взаимосвязей объектов, фиксируется архитектурой прикладного программного приложения, прежде всего, зависит от операционной системы, в которой разрабатывалось прикладное приложение.

Освоение материала пятого сегмента, предполагает, что обучаемый вышел на достаточно высокий профессиональный уровень. В данном случае, очень важным является самостоятельный анализ различных открытых источников.

Сегмент шестой. Управление работой прикладного программного продукта Microsoft Office с использованием сгенерированных конструкций VBA.

Формирование знаний шестого сегмента позволяет эффективно использовать возможности генерации программного кода, которые реализованы в прикладных программных продуктах Microsoft Office. Конструкции встроенного языка прикладных продуктов Microsoft Office могут генерироваться автоматически. Сгенерированные макросы могут включать любую последовательность действий пользователя. Таким образом, пользователи прикладных продуктов Microsoft Office имеют возможность автоматически создавать макросы. В этом случае, пользователи могут пользоваться языком VBA, не изучая используемых конструкций языка.

Важной особенностью является то, что сгенерированные конструкции доступны для редактирования. Таким образом, пользователь может управлять работой сгенерированных в макросе команд. В связи с тем, что VBA является объектно-ориентированной системой, сгене-

рированные конструкции содержат объекты. Для управления объектами необходимо знание программирования на профессиональном уровне. В том случае, если сгенерированный макрос редактирует профессиональный программист, то возможности управления работой прикладной системы с использованием сгенерированных конструкций в значительной степени возрастают [10].

Материал шестого сегмента проработан далеко не во всех открытых источниках. Следовательно, упор необходимо делать на самостоятельную работу и необходимые консультации.

Сегмент седьмой. Программная обработка файлов.

Седьмой сегмент, включает много важной информации для работы с внешними носителями. Экономическая информация характеризуется большими объемами данных. Сформированные знания по использованию внешних носи-

телей могут оказаться полезными для многих пользователей при работе с большими объемами информации.

Работу с файлами в VBA подразделяется на два основных направления:

Во-первых, доступ к файлам;

Во-вторых, управление файлами.

При изучении доступа к файлам рассматривается программная обработка, предназначенная для занесения информации в файл или чтение информации из файлов. Управление файлами включает такие действия, как создание и удаление папок, просмотр папок, перемещение файлов, удаление файлов и другие.

Использование файловой обработки в программной среде VBA позволяет обеспечить широкие возможности обмена информацией конкретного приложения с другими прикладными программными продуктами [11].

Данный вид программной обработки, как правило, не вызывает за-

труднений. Поэтому для изучения может быть использован любой открытый ресурс.

В заключении хотелось бы обратить внимание на географию российских вузов, в которых преподается дисциплина «Офисное программирование»: Национальный открытый университет ИНТУИТ (Москва), Омский Государственный технический университет, Тверской университет, Поволжский государственный университет (Самара), Национальный исследовательский Томский политехнический университет и многие другие вузы РФ. Во многих, вузах выкладываются в открытый доступ некоторые методические материалы.

По-видимому, переход к информационному обществу требует дальнейшего развития преподавания дисциплины «Офисное программирование» уделяя должное внимание возможностям открытого образования.

## Литература

1. *Hornsby, Peter* (2009-08-03). «Empowering Users to Create Their Own Software». URL: <http://www.uxmatters.com/mt/archives/2009/08/empowering-users-to-create-their-own-software.php> (дата обращения: 31.01.2014).
2. *Scaffidi, Christopher; Brandt, Joel; Burnett, Margaret; Dove, Andrew; Myers, Brad* (2012). «SIG: end-user programming» // CHI '12 Extended Abstracts on Human Factors in Computing Systems: pp. 1193–1996. URL: <http://dl.acm.org/citation.cfm?id=2212776.2212421&coll=DL&dl=GUIDE> (дата обращения: 25.07.2015).
3. *Sarkar, Advait; Blackwell, Alan; Jamnik, Mateja; Spott, Martin*. «Teach and Try: A simple interaction technique for exploratory data modelling by end users». URL: [http://www.cl.cam.ac.uk/~as2006/files/sarkar\\_2014\\_teach\\_try.pdf](http://www.cl.cam.ac.uk/~as2006/files/sarkar_2014_teach_try.pdf) (дата обращения: 15.10.2015).
4. *Рычкова А.В., Смирнов А.А.* Методические аспекты повышения эффективности обучения в smart-университете // Открытое образование. 2015. № 5. С. 39–43.
5. *Margaret M. Burnett and Christopher Scaffidi*. End-User Development // Interaction-Design.org.at «Encyclopedia of Human-Computer Interaction». URL: <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/end-user-development> (дата обращения: 03.11.2015).
6. *Тельной В.И., Рычкова А.В.* Применение трехмерного моделирования при чтении лекций по начертательной геометрии // Вестник МГСУ. – 2014. – № 5. С. 176 – 183.
7. *Abraham, R.; Burnett, M; Erwig, M* (2009). Spreadsheet programming// Wiley Encyclopedia of Computer Science and Engineering. pp. 1–10
8. *Lieberman, H., Paternò, F., Klann, M., and Wulf, V.* (2006). End-User Development: An Emerging Paradigm // In: End-User Development, Lieberman, H., Paternò, F., and Wulf, V. (eds.), Springer Netherlands, 2006, ser. Human-Computer Interaction Series, vol. 9, Chapter 1, pp. 1–7. URL: [http://link.springer.com/chapter/10.1007%2F1-4020-5386-X\\_1](http://link.springer.com/chapter/10.1007%2F1-4020-5386-X_1) (дата обращения: 13.11.2015).
9. *Смирнов А.А.* Реализация основных принципов smart образования при обучении в магистратуре // В сборнике: Ценности и интересы современного общества Информационные технологии. Материалы конференции. 2014. С. 147 – 152.
10. *Maria Francesca Costabile, Daniela Fogli, Piero Mussio, Antonio Piccinno*. End-user development: the software shaping workshop approach // In Lieberman, H., Paternò, F., Wulf, V. (Eds) (2004) End User Development – Empowering People to Flexibly Employ Advanced Information and Communication Technology, © 2004 Kluwer Academic Publishers, Dordrecht, The Netherlands.
11. *Roy Chowdhury, Soudip; Rodriguez, Carlos; Daniel, Florian; Casati, Fabio* (2010). Wisdom-aware computing: on the interactive recommendation of composition knowledge. URL: <http://dl.acm.org/citation.cfm?id=1987701> (дата обращения: 22.12.2015).