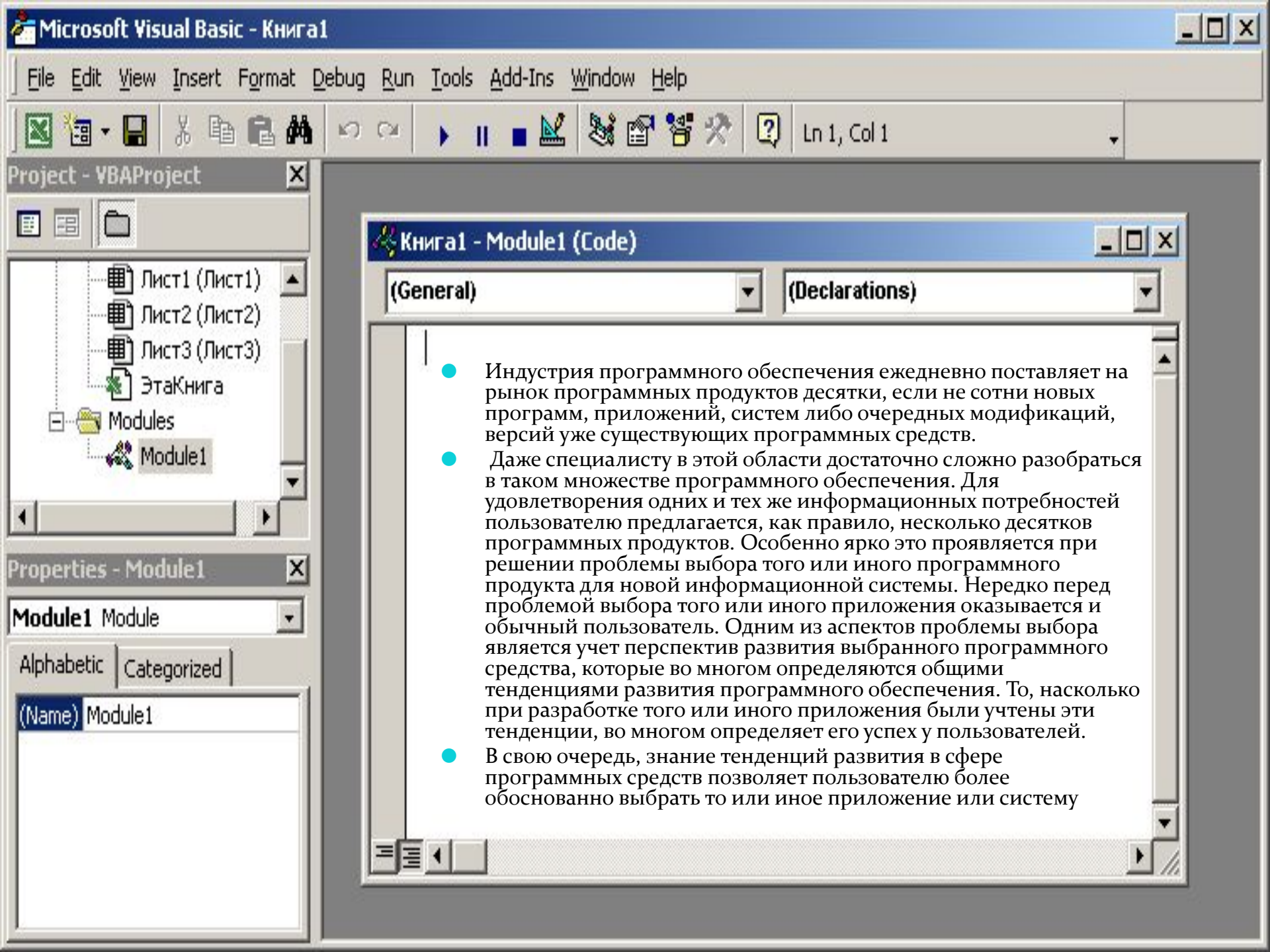


```
each: function(e, t, n) {
  var r, i = 0,
      o = e.length,
      a = N(e);
  if (n) {
    if (a) {
      for (; o > i; i++)
        if (r = t.apply(e[i], n), r !== !1) break;
    } else
      for (i in e)

```

Программирование в сфере офисных приложениях. Тенденции развития программного обеспечения

```
return null == e ? "" : b.call(e);
} : function(e) {
  return null == e ? "" : (e + "").replace(C, "");
},
makeArray: function(e, t) {
  var n = t || [];
  return null != e && (N(Object(e)) ? x.merge(n, "string" == typeof e ? [e] : e) : h.call(n, e)), n
},
isArray: function(e, t, n) {
  var r;
  if (t) {
    if (e) return e.call(t, e, n);
    for (r = t.length, r = n ? 0 > n ? Math.max(0, r + n) : 0 : 0; r > n; r++)
      if (n in t && t[r] === e) return n;
  }
}
```



Project Explorer showing a tree view with folders and files: Лист1 (Лист1), Лист2 (Лист2), Лист3 (Лист3), ЭтаКнига, Modules, and Module1.

Properties window for Module1, showing the name "Module1" and tabs for "Alphabetic" and "Categorized".

Книга1 - Module1 (Code)

(General) (Declarations)

- Индустрия программного обеспечения ежедневно поставляет на рынок программных продуктов десятки, если не сотни новых программ, приложений, систем либо очередных модификаций, версий уже существующих программных средств.
- Даже специалисту в этой области достаточно сложно разобраться в таком множестве программного обеспечения. Для удовлетворения одних и тех же информационных потребностей пользователю предлагается, как правило, несколько десятков программных продуктов. Особенно ярко это проявляется при решении проблемы выбора того или иного программного продукта для новой информационной системы. Нередко перед проблемой выбора того или иного приложения оказывается и обычный пользователь. Одним из аспектов проблемы выбора является учет перспектив развития выбранного программного средства, которые во многом определяются общими тенденциями развития программного обеспечения. То, насколько при разработке того или иного приложения были учтены эти тенденции, во многом определяет его успех у пользователей.
- В свою очередь, знание тенденций развития в сфере программных средств позволяет пользователю более обоснованно выбрать то или иное приложение или систему



- Офисное программирование — это процесс разработки приложений, предназначенных для автоматизации офисной деятельности с использованием специализированных пакетов (MS Office, OpenOffice.org или подобных).
- Офисное программирование как процесс разработки приложений имеет ряд особенностей по сравнению с программированием как таковым. В основном они проявляются в таких аспектах, как:
 - цель разработки;
 - область применения;
 - язык программирования;
 - среда разработки;
 - поддержка объектно-ориентированного программирования.
- Рассмотрим эти особенности на примере MS Office.
-

1 ЦЕЛЬ РАЗРАБОТКИ

В офисной среде программный проект неразрывно связан с документом, хранится как часть документа и не может существовать независимо от него. Документ, а не программа, является целью разработки.

Стандартные возможности среды по работе с документами велики. Однако возможность изменить типовой документ, снабдив его дополнительными функциями – это одна из важнейших задач офисного программирования. Для ее решения офисная среда представляет совокупность библиотек классов, которые составляют каркас (Framework) текстовых документов, электронных таблиц, презентаций, баз данных и приложений на основе этих документов. Всякий раз, когда создается новый документ, его каркас составляют объекты библиотек, заданные по умолчанию. Этот каркас можно существенно изменить, добавив в документ новые свойства. Расширение каркаса не требует от программиста значительных усилий – достаточно включить в него необходимые библиотеки классов.

ОБЛАСТЬ ПРИМЕНЕНИЯ

Область применения офисного программирования весьма и весьма широка – от настройки отдельных документов до решения задач автоматизации офисной деятельности масштаба предприятия, в т.ч. ориентированных на совместную работу в глобальной сети.

Понятно, что основное назначение офисных приложений — автоматизация офисной деятельности, однако, средства офисного программирования применяют и для совсем других разработок, выходящих далеко за ее рамки. Для наглядности приведем ряд примеров: программа биоритмов, написанная на VBA для Excel; ASCII art на музыкальный клип AC/DC; макровирусы и множество других совсем не офисных приложений.



ЯЗЫК ПРОГРАММИРОВАНИЯ

Единственный язык программирования, поддерживаемый пакетом MS Office является [Visual Basic for Application](#) (VBA). VBA — это инструмент разработки приложений, который позволяет создавать программные продукты, решающие практически все задачи, встречающиеся в среде MS Windows. В случае офисных приложений, программы или макросы на VBA можно использовать, например, для оформления документов (подготовки текстов) или анализа данных таблиц (электронных таблиц).

VBA – уникальное приложение, поскольку оно встраивается в другое приложение и расширяет его функциональные возможности. С помощью VBA можно:

- создать собственное диалоговое окно и придать ему требуемый внешний вид;

- создать макросы, расширяющие функциональные возможности приложения, в которое встроено VBA;

- изменить меню приложения Microsoft Office;

- управлять другим приложением Microsoft Office или принадлежащими ему данными;

- объединить данные из нескольких приложений Microsoft Office в одном документе;

- автоматически создавать или изменять страницы Web, совместно используя приложения Microsoft Office и VBA.

Для разработчика доступны следующие инструменты и средства, которые используются при создании проекта VBA:

- отладка приложений без предварительной компиляции;

- средства Win32 API;

- SQL и объекты доступа к данным для управления данными и извлечения их из внешних источников данных, таких как Microsoft SQL Server;

- построение и проверка элементов интерфейса непосредственно в среде разработки VBA (Integrated Development Environment, IDE);

- связывание программ и процедур с событиями, которые возникают в приложениях VBA.

- СРЕДА РАЗРАБОТКИ
- Среда приложений Office ориентирована в первую очередь на пользователей, а не на программистов и в ней можно создавать документы без всякого программирования. Поэтому программист обычно начинает работать с документами не на пустом месте, а с их заготовками, созданными пользователями, т.е. и сам программист может выступать в роли пользователя. Средства совместной работы над документами Office обеспечивают одновременную работу программистов и пользователей.
- Среда MS Office предлагает два способа создания программ, отличающихся подходом к процессу: использование макрорекордера и написание исходного кода программ на языке VBA в интегрированной среде разработки. Эти подходы ориентированы на разные категории: непосредственно пользователей и программистов соответственно.
- Макрорекордер (MacroRecorder) – это программный инструмент, записывающий действия пользователя при работе с документами и приложениями, с сохранением записи в виде макроса — исходного кода на языке VBA. При вызове сохраненного макроса воспроизводится вся сохраненная последовательность действий.
- Макрорекордер представляет возможность создания программного проекта или, по крайней мере, его отдельных компонентов автоматически, без программирования. Для записи и воспроизведения макроса не требуется специальных знаний, поэтому пользователь может самостоятельно создавать программы (макросы), в общем случае даже не представляя себе как они работают.
- Для программиста макрорекордер полезен тем, что позволяет создавать фрагменты программы автоматически, тем самым увеличивая скорость разработки и уменьшая время отладки.
- Интегрированная среда разработки на VBA (Visual Basic Environment, VBE) — встроенное в MS Office средство для написания, тестирования и отладки приложений на VBA. Среда VBE представляет все возможности для создания законченных офисных приложений, включая средства визуального проектирования пользовательского интерфейса. VBE ориентирована на использование программистами для разработки офисных приложений (это отнюдь не означает, что пользователи не могут применять VBE).

ПОДДЕРЖКА ООП

- Разработка приложений для MS Office тесно связана с парадигмой объектно-ориентированного программирования. Все документы (более того, сами компоненты пакета) в MS Office — суть объекты, наделенные собственными наборами свойств (характеристик объекта), методов (подпрограмм управления свойствами) и событий (подпрограмм, обрабатывающих изменения состояния объекта в результате некоторых действий). Соответственно, для обеспечения более полной интеграции с пакетом, входной язык (VBA) также поддерживает ООП. Все объекты приложения MS Office образуют иерархическую структуру, которая определяет связь между ними и способ доступа. Такая структура называется объектной моделью (object model).
- За рамки объектной модели выходят, но также могут использоваться в офисных приложениях, внешние объекты, поддерживающие технологии DDE, OLE/ActiveX и ряд других. В объектно-ориентированную концепцию удачно вписывается технология визуального программирования. Все отображаемые элементы графического интерфейса, такие как формы, элементы управления, меню и панели инструментов являются объектами, наделенными набором свойств и методов и способными реагировать на события (например, щелчки мыши, нажатия клавиш и т.п.). При визуальном подходе не требуется программного задания (хотя это и возможно) их основных свойств (например, ширина или высота, цвет фона и т.п.). Эти свойства можно задать при помощи мыши (например, ширину и высоту формы путем операции «перетаскивания» маркеров) или установить их в окне свойств (название формы, цвет фона формы и т. д.).
- Таким образом, визуальное программирование делает проектирование интерфейса программы более наглядным и быстрым. При этом сохраняется возможность управлять всеми объектами и программно.

1.6 ПРЕИМУЩЕСТВА ОФИСНОГО ПРОГРАММИРОВАНИЯ

- Преимущества, которые получает конечный пользователь, использующий программируемые офисные документы:
- Пользователь получает документы, обладающие новыми функциями и способные решать задачи, характерные для проблемной области пользователя.
- Пользователь находится в единой офисной среде независимо от того, с каким документом он работает в данный момент и какой программист разрабатывал этот документ.
- Большинство доступных при работе с документами функций являются общими для всех документов, поскольку их предоставляет сама офисная среда. Единый стиль интерфейса разных документов облегчает работу с ними.
- Пользователь сам, не будучи программистом, способен создавать простые виды программируемых офисных документов, постепенно совершенствуясь в этой деятельности.
- Преимущества, которые получает программист, работающий в office:
- В распоряжении программиста находится мощная интегрированная среда. Для него эта среда представлена в виде совокупности хорошо организованных объектов, доступных в

ОСНОВНЫМИ ТЕНДЕНЦИЯМИ РАЗВИТИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Основными тенденциями развития программного обеспечения являются:

— стандартизация как отдельных компонентов программных средств, так и интерфейсов между ними, которая позволяет использовать то или иное приложение на разных аппаратных платформах и в среде разных операционных систем, а также обеспечить его взаимодействие с широким кругом приложений;

— ориентация на объектно-ориентированное проектирование и программирование программных средств, что позволяет в совокупности с их стандартизацией перейти к новой технологии — технологии «сборки» того или иного приложения, ориентированного на конкретные потребности конкретного пользователя, из отдельных модулей-«кубиков», избегая тем самым ненужных пользователю функций.

При этом снижаются объем и стоимость и повышается надежность «собранного» таким образом приложения (существующие офисные системы, например, используются абсолютным большинством обычных пользователей не более чем на 20 — 30%). Одновременно во многом снимается проблема модификации приложения при изменении информационных потребностей пользователя. Кроме того, ориентация на технологии «сборки» приложения в сочетании с возможностями сетевого доступа к этим приложениям как требуемым программным ресурсам позволяет в перспективе не приобретать в «личное пользование» то или иное приложение, а брать его во временную «аренду», снижая тем самым затраты на программное обеспечение;

— интеллектуализация интерфейса пользователя, обеспечение его интуитивной понятности, непроцедурности и приближение языка общения с компьютером к профессиональному языку пользователя; настройка интерфейса пользователя на особенности и потребности конкретного пользователя при организации его диалога с компьютером; использование средств мультимедиа при реализации интерфейса пользователя;

— интеллектуализация возможностей программ и программных систем; все шире при проектировании приложений используются методы искусственного интеллекта, что позволяет сделать приложения более «умными» и решать все более сложные, плохо формализуемые задачи;

— универсализация отдельных компонентов (модулей) прикладных программ и постепенный переход этих компонентов, а затем и самих программ из области специализированного прикладного ПО в область универсального прикладного ПО.



Реализация информационных процессов требует соответствующего программного обеспечения (программных средств), затраты на приобретение которого соизмеримы с затратами на аппаратное обеспечение.

Различают три вида ПО: системное, прикладное и инструментальное. В основе классификации и соответственно определения этих видов ПО положена инвариантность (независимость) ПО по отношению к специфике той или иной проблемной области, в которой предполагается его использование.

Типичным представителем системного ПО являются операционные системы (ОС), обеспечивающие эффективное использование аппаратно-программных ресурсов ЭВМ и предоставление пользователю дружественного интерфейса для работы на компьютере. Пользователь взаимодействует с программами только через «посредничество» ОС. Современные ОС, как правило, являются сетевыми.

Прикладное программное обеспечение делится на универсальное и специализированное. Типичными представителями универсального прикладного ПО являются офисные системы, системы управления базами данных и системы искусственного интеллекта.

Офисная система — совокупность приложений, обеспечивающих автоматизацию типичных функций организационного управления деятельностью офиса. Наиболее часто используемым приложением офисной системы являются текстовый и табличный процессоры. В офисной системе MS Office это соответственно MS Word и MS Excel.

Данные, требующиеся для решения задач пользователя, структурируются и накапливаются в базах данных (БД). Структуру данных в БД для представления той или иной предметной области определяет модель данных. Выделяют четыре основных типа моделей данных: иерархическую, сетевую, реляционную и объектную. Для задачи создания, пополнения, модификации и поиска данных в БД используют систему управления базой данных (СУБД). При работе с БД в среде некоторой СУБД различают три уровня представления данных: концептуальный, внешний и внутренний. Пользователь работает с данными в среде СУБД на внешнем уровне представления данных. Наиболее важная из функций СУБД — предоставление в распоряжение пользователя языка запросов, посредством которого он выражает свои информационные потребности.

Перспективным направлением является искусственный интеллект (ИИ) как совокупность средств и методов решения интеллектуальных задач на основе моделирования процессов выявления, представления и манипулирования знаниями с использованием средств вычислительной техники.

Для автоматизации процессов разработки программного обеспечения предназначены инструментальные системы, к которым относятся системы программирования и CASE-системы