**Курсовая работа по информатике.**

***Тема***: *“Разработка программированных средств с помощью VBA”.*

2002

***Оглавление:***

Ведение.

1. Принципы информационных технологий обучения.

# 1.1. Компьютерное тестирование.

1. Краткая история языка VBA.
2. Зачем изучать язык VBA.
3. Использование функций Excel.

**5.** Создание настраиваемых диалоговых окон:

5.1.Отображение форм с помощью VBA.

5.2.Использование VBA с элементами управления в форме.

1. Список литературы.

# Введение

Моя тема актуальна, потому что на данный момент очень большое количество высших и средних учебных заведений используют программированные средства в обучении, в принятии экзаменов. Программированное обучение, возникшее в начале 60-х гг. на основе новых дидактических, психологических, и кибернетических идей. ПО направило свои усилия на создание такой технологии учебного процесса, которая позволяла бы контролировать каждый шаг продвижения учащегося по пути познания и благодаря этому оказывать ему своевременную помощь, избавляя тем самым от многих затруднений, потери интереса и других негативных последствий, сопровождающих плохо управляемый процесс.

Современные обучающие машины быстро устанавливают уровень обученности и возможности работающих с ними учеников, могут “приспосабливаться” к ним. Современные обучающие программы чаще всего составляются по смешанной схеме, что позволяет сделать их гибкими.

# Принципы информационных технологий обучения

С началом промышленного изготовления компьютеров первых поколений и их появлением в образовательных учреждениях возникло новое направление в педагогике – компьютерные технологии обучения. По-настоящему массовыми создание и использование обучающих программ стали с начала 80-х годов, когда появились и получили широкое распространение персональные компьютеры. С тех пор образовательные применения ЭВМ выдвинулись в число их основных применений наряду с обработкой текстов и графиков, оттеснив на второй план математические расчеты.

С появлением примеров компьютерного обучения к созданию компьютерных обучающих программ приобщились десятки тысяч педагогов – специалистов в различных областях знаний, чаще всего в технологических науках. В разрабатываемых ими программах, опираясь в основном на интуицию и практический опыт, они воплощали свои представления конкретных дисциплин с помощью компьютеров. Педагоги-теоретики долгое время оставались в стороне от этого нового направления в обучении. В результате до сих пор отсутствует общепризнанная психолого-педагогическая теория компьютерного обучения, компьютерные обучающие программы продолжают создаваться и применяться без необходимого учета принципов и закономерностей обучения.

Благодаря своим конструктивным и функциональным особенностям современный персональный компьютер является уникальной по своим возможностям обучающей машиной. Он находит применение в обучении самым разнообразным дисциплинам и служит базой для создания большого числа новых информационных технологий обучения.

Технологические возможности персонального компьютера, если компьютер используется как обучающее средство, позволяют:

* Активизировать учебный процесс;
* Индивидуализировать обучение;
* Повысить наглядность в предъявлении материала;
* Сместить акценты от теоретических знаний к практике;
* Повысить интерес учеников к обучению.

Активизация обучения связана с диалоговым характером работы компьютера и с тем, что каждый ученик работает за своим компьютером. При традиционном классном обучении основное – это восприятие учащимися информации в устной форме, при этом ученику не часто приходится проявлять активность на уроке и учитель не в состоянии организовать и контролировать активную работу каждого ученика на его рабочем месте.

Поэтому традиционное обучение, в основном, является пассивным – многие педагоги сетуют, что на уроке активно работают 20 – 30% учащихся. Если же обучение ведется в компьютерном классе, компьютер диалоговым характером своей работы стимулирует ученика к деятельности и контролирует ее результаты.

Нельзя рассматривать компьютер в обучении отдельно, сам по себе, в отрыве от:

* Программного обеспечения – педагогических программных средств;
* Организационных форм использования компьютеров,

В настоящее время существует огромное множество обучающих программ по самым различным предметам, ориентированных на самые различные категории учащихся, начиная контингентом детских садов и кончая персоналом атомных электростанций. Кроме того, каждая из программ предназначена только для одного типа компьютеров – а ведь этих типов великое множество – и не годится для других!

# Компьютерное тестирование

Обучение – многогранный процесс, и контроль знаний – лишь одна из его сторон. Однако именно в ней компьютерные технологии продвинулись максимально далеко, и среди них тестирование занимает ведущую роль. В ряде стран тестирование потеснило традиционные формы контроля – устные письменные экзамены и собеседования.

По- видимому, многие преподаватели уже прошли через некоторую эйфорию при создании тестов и поняли, что это- весьма непростое дело. Куча бессистемно надерганных вопросов и ответов- далеко еще не тест. Оказывается, что для создания адекватного и эффективного теста надо затратить много труда. Компьютер может оказать в этом деле немалую помощь.

Существует специальная теория тестирования, оперирующая понятиями надежность, валидность, матрица покрытия и т.д., не специфических именно для компьютерных тестов. Здесь мы не будем в нее углубляться, сосредоточившись в основном на технологических аспектах.

Как отмечалось выше, широкое распространение в настоящее время получают инструментальные авторские системы по созданию педагогических средств: обучающих программ, электронных учебников, компьютерных тестов. Особую актуальность для преподавателей школ и вузов приобретают программы для создания компьютерных тестов – тестовые оболочки. Подобных программных средств существует множество, и программисты – разработчики готовы строить новые варианты, так называемых, авторских систем. Однако широкое распространение этих программных средств сдерживается отсутствием простых и нетрудоемких методик составления тестовых заданий, с помощью которых можно «начинять» оболочки.

***Типы компьютерных тестов.*** В соответствии с моделью занятий выделим три класса компьютерных тестов на задания, умения и навыки. Отметим, что типы компьютерных тесовых заданий определяются способами однозначного распознавания ответных действий тестируемого.

1. Типы тестовых заданий по блоку «знаний»:
   * Вопросы альтернативные (требуют ответа да – нет);
   * Вопросы с выбором (ответ из набора вариантов);
   * Вопросы информативные на знание фактов (где, когда, сколько);
   * Вопросы на знание фактов, имеющих формализованную структуру (в виде информационной модели или схемы знаний);
   * Вопросы по темам, где имеются однозначные общепринятые знаковые модели: математические формулы, законы, таблицы;
   * Вопросы, ответы на которые можно контролировать по набору ключевых слов;
   * Вопросы, ответы на которые можно распознавать каким-либо методом однозначно.
2. Типы тестовых заданий по блоку «навыки»:

* Задания на стандартные алгоритмы (альтернативные да – нет, выбор из набора вариантов);
* выполнение действия.

1. Типы тестовых заданий по блоку «умения». Те же самые, что навыки, но использующие нестандартные алгоритмы и задачи предметной области при контроле времени их решения:

* Задания на нестандартные алгоритмы (альтернативные да – нет, выбор из набора вариантов);
* Выполнение действий.

Выбор типов тестов определяется

* Особенностями инструментальных тестовых программ (тестовыми оболочками);
* Особенностями предметной области;
* Опытом и мастерством экспертов.

# Краткая история языка VBA

Несмотря на новизну языка Visual Basic for Applications, история его проявления почти так же стара, как и вся компьютерная промышленность. Про язык VBA можно сказать, что он является диалектом языка BASIC, который появился в начале 60-х.

Хотя по сегодняшним понятиям язык BASIC был довольно ограниченным и, как теперь говорят, варварским, он был прост для изучения и очень скоро получил широкое распространение. Версии BASIC выпускался для всех типов компьютеров. Язык GWBASIC производство компании Microsoft был одним из первых языков программирования для современных персональных компьютеров. Он поставлялся со всеми операционными системами MS DOS до 5-й версии. Ранние персональные компьютеры производства компании IBM даже имели версию BASIC, встроенную в ПЗУ.

С годами первоначальная версия BASIC была существенно доработана. Менялась технология программирования, и вместе с ней под влиянием разработчиков программного обеспечения менялся сам BASIC. Современный его диалект включает многие черты и свойства, характерные для более поздних и совершенных языков, таких как Pascal, C и C++.

В конце 80-х Microsoft выпускает существенно улучшенную версию BASIC, названную QuickBASIC во все версии MS DOS, начиная с 6-й (но не Windows 95).

После нескольких версий QuickBasic в 1992 году Microsoft выпускает Visual Basic for Windows.

Язык VBA в основном совпадает с Visual Basic for Windows, но имеет и существенное отличие. В частности, макросы VBA хранятся в файле документа того приложения, в котором вы создаете этот макрос.

Внедрив один язык макросов во все свои приложения, Microsoft гарантирует, что большая часть того, что вы выучите о VBA применительно к одному приложению, будет справедлива и для остальных.

***Зачем изучать язык VBA.***

На первый взгляд может показаться, что если вы можете записать макрос средствами программы Excel и потом воспроизвести его, то вам совсем не нужно учить язык VBA. Однако это не так. Записанный макрос не универсален, поскольку он может только повторить однажды выполненную последовательность команд, и только в том же самом порядке, в котором вы ее записали. С помощью VBA можно создать макрос, который будет проверять некоторые заданные вами условия и в зависимости от них менять последовательность или состав выполняемых команд.

Например, вы пытаетесь выполнить в программе Excel макрос, который должен открыть лист с названием ***Продажи***. Если такого листа в текущей книге нет, макрос будет остановлен и Excel выдаст сообщение об ошибке. Но если вы отредактируете свой макрос с помощью VBA, он может сначала проверить наличие нужного листа, а в случае его отсутствия может даже его создать.

Если вам нужно циклически повторить несколько раз некоторую последовательность команд, записанный макрос - плохой помощник. Ведь при выполнении он повторит эту последовательность ровно столько раз, сколько раз вы повторили ее при записи - по крайней мере, до тех пор, пока вы не отредактируете его или не перезапишите.

Другое дело VBA - макрос, который может проверить заданное вами условие или просто перед выполнением запросить информацию о количестве повторений и о том, нужно ли вообще выполнять то или иное действие.

Представьте себе, что вы записали макрос, который меняет ширину столбцов в листе Excel. Если вам требуется, чтобы он менял ширину всех трех столбцов вручную. Записанный макрос всегда будет менять ширину столбцов со второго по четвертый. Отредактировав записанный макрос с помощью VBA, вы добьетесь того, чтобы он спрашивал вас, сколько столбцов нужно обработать и какие именно. Вы даже сможете задать новую ширину.

Это только самый простой пример того, что вы можете делать с макросом с помощью VBA. На самом деле существует очень много обстоятельств, требующих проверки условия для принятия решения или выполнения циклических операций. И единственный способ сделать это - включить в записанный макрос операторы VBA.

Помимо редактирования записанных макросов, вы можете применять VBA для согласованного управления несколькими простыми макросами, каждый из которых выполняет некоторую часть большого задания.

Например, вы можете регулярно импортировать данные из базы данных в лист Excel, формировать, эти данные, строить на их основании диаграмму, а затем посылать и диаграмму, и отформатированный отчет на печать.

Для того чтобы собрать все эти, вполне независимые задания, в одно, и оформить их в виде макроса, вам понадобится записать по макросу для каждого задания - макроса для импорта данных, макрос для форматирования, макрос для построения диаграммы и для вывода на печать. Потом вы сможете запускать эти макросы на выполнение с помощью VBA в нужной последовательности.

Кроме того, с помощью VBA вы можете управлять выполнением других программ, воспользовавшись таким средством, как автоматизация, и сможете организовать обмен данными с помощью OLE.

***Использование функций Excel***

В дополнение к встроенным функциям Visual Basic for Applications, приложение Excel обладает широким разнообразием функций для выполнения математических, логических, финансовых и статистических операций над данными в рабочих листах. Многие из этих функций доступны для использования в VBA.

Функции, которые Excel позволяет использовать в VBA, не является частью VBA, они входят в состав этого приложения. Функции, которые можно использовать в одном приложении, могут быть недоступными в другом. При написании процедур VBA для работ в любом приложении не следует использовать такие функции, потому что они могут быть недоступными в другом приложении.

Для того чтобы использовать функции, которые принадлежат какому-либо приложению, нужно получить доступ к функции в среде VBA через объект Application. Этот объект представляет приложение и все его ресурсы.

Не всякая функция Excel доступна к применению в среде VBA. Для того чтобы определить, какие из функций Excel доступны, как правильно их использовать и для чего они предназначены, воспользуйтесь справочной системой по ключевому слову *function*.

***Создание настраиваемых диалоговых окон.***

VBA позволяет создавать и использовать в программах нестандартные (настраиваемые) диалоговые окна, добавляя объект *UserForm* в проект. Используя разрабатываемые пользователем формы VBA, можно создавать нестандартные диалоговые окна для отображения данных или получения значений от пользователя программы в том виде, который наиболее соответствует потребностям программы. Например, можно создать тест, отобразить диалоговое окно для отображения вопросов с вариантами ответов и предоставить пользователю возможность выбрать один из вариантов ответа, который он считает верным.

Нестандартные диалоговые окна позволяют программе взаимодействовать с её пользователем самым сложным образом и обеспечивают разнообразную форму ввода и вывода данных.

Нестандартное диалоговое окно создаётся в VBA посредством добавления объекта *UserForm* в проект. Этот объект представляет собой пустое диалоговое окно; оно имеет строку заголовка и кнопку закрытия, но в нём отсутствуют какие-либо другие элементы управления. Нестандартное диалоговое окно создаётся путем добавления элементов управления в объект *UserForm* (обычно называемый просто *формой*). Каждый объект *UserForm* имеет свойства, методы и события, наследуемые им от класса объектов *UserForm.*

Каждый объект *User Form* также содержит модуль класса, в который пользователь добавляет собственные методы и свойства или вписывает процедуры обработки событий для данной формы.

Для того чтобы создать собственный интерфейс, независимый от среды Excel, необходимы экранные формы.

*Экранные формы -* это окна различного назначения и вида, созданные пользователем для своего приложения. Они содержат элементы управления, позволяющие пользователю обмениваться информацией с приложением.

При создании нового объекта *UserForm* в проекте создаётся новый подкласс объекта *UserForm.*

Любые процедуры или функции, записываемые в разделе *General* модуля класса формы, становятся дополнительными методами подкласса данного конкретного объекта. Новые свойства для формы могут быть также созданы посредством добавления процедур *Propery Let* в её модуль класса. Процедуры добавляются в наследуемый модуль класса. Новые экземпляры подкласса *UserForm* создаются с помощью инструкции *Dim* и ключевого слова *New.* Однако в большинстве случаев управление формой будет осуществляться посредством использования стандартных методов и свойств класса *User Form* и создания процедур обработки событий для конкретной формы и её элементов управления.

Кроме методов, свойств и событий, которые объект *UserForm* представляет изначально, VBA предоставляет две инструкции, которые особенно полезны при работе с объектами форм: *Load* и  *Unload.* Эти инструкции можно использовать для загрузки формы в память или для удаления её из памяти.

Объект *User Form* может содержать элементы управления подобные тем, что находятся в других диалоговых окнах, отображаемых Excel и другими Windows-приложениями. *Элементы управления* - это элементы диалогового окна, которые позволяют пользователю взаимодействовать с программой. К этим элементам относят кнопки переключателей, текстовые поля, линейки прокрутки, командные кнопки и т.п.

Каждый элемент управления - объект со специальными свойствами, методами и событиями. Подобно содержащим их формам, средства элементов управления можно определять программным путём или с помощью окна Properties редактора Visual Basic. Значения свойства элемента управления присваиваются или получаются в VBA- программе, так же, как и для любого другого объекта.

***Отображение форм с помощью VBA.***

VBA использует созданный графический дизайн формы - с настройками свойств формы и элементов управления - для получения всей информации, необходимой для отображения диалогового окна: размеров диалогового окна, элементов управления в нём и т.п. В результате VBA позволяет отобразить форму диалогового окна с помощью единственной инструкции.

Для отображения нестандартного диалогового окна используется метод *Show* объекта *UserForm.*

Если в настоящий момент форма не загружена в память, метод *Show* загружает форму и отображает её. Если форма уже загружена, метод *Show* просто отображает её. В любом случае этот метод отображает форму, а затем присваивает ей фокус. Форма остается до тех пор, пока не будет выполнен метод *Hide* объекта *UserForm,* либо пока форма не будет выгружена с помощью инструкции *Unload.*

Форма остаётся загруженной до тех пор, пока экземпляр формы не выйдет за пределы видимости, - т.е. процедура, создавшая этот экземпляр объекта формы, перестанет выполняться - либо пока форма не будет выгружена с помощью инструкции *Unload.*

Хотя программа в модуле класса формы будет выполняться в результате события в диалоговом окне, общее выполнение программы приостанавливается до тех пор, пока форма диалогового окна не будет закрыта или скрыта.

***Использование VBA с элементами управления в форме.***

Отображения одного диалогового окна для выполнения задачи обычно недостаточно. Почти всегда требуется определить состояние элементов управления диалогового окна с целью выяснить, какие данные или опции выбрал пользователь. Например, если диалоговое окно используется для получения от пользователя информации о том, по каким столбцам и строкам должно выполняться упорядочение рабочего листа, необходимо иметь возможность выяснить, какие значения пользователь ввел после закрытия диалогового окна и до действительного начала операции упорядочивания.

В других случаях может потребоваться динамическое изменение заголовков кнопок (или других элементов управления) диалогового окна, динамическое обновление надписи или поля, связанного со счетчиком, или динамическое подтверждение введенных в диалоговое окно данных.

***Список литературы:***

1. Свириденко С.С. Современные информационные технологии. – М.: Радио и связь,1989.
2. Советов Б.Я. АСУ. Введение в специальность. – М.: Высшая школа,1989.
3. Советов Б.Я. Информационная технология. – М.: Высшая школа, 1992.
4. Фурунжиев Р.И., Гугля В.А. САПР, или как ЭВМ помогает конструктору. – Минск: Высшая школа, 1987.
5. Журнал «Информатика и образование», с 1992г.
6. Журнал «Педагогическая информатика», с 1994г.
7. Белошапка В.К. Информационное моделирование. – Омск: Изд-во ОГПИ, 1992.
8. Гнеденко Б.В., Коваленко И.Н. Введение в теорию массового обслуживания. – М.: Наука, 1966.
9. Косневски Ч. Занимательная математика и персональный компьютер. – М.: Просвещение, 1989.
10. Липаев В.В. Проектирование программных средств. – М.: Высшая школа, 1990.
11. Пидкасистый П.И. педагогика. – М.: Педагогическое общество Россия,2000.
12. Ковальски С. Excel 2000 без проблем. – М.: Бином, 2000.
13. Харис М. Программирование для Microsoft Excel 2000 за 21 день. – М.: Вильямс, 2000.
14. Могилев А.В., Пак Н.И., Хеннер Е.К. Информатика. – М.: ACADEMIA, 2000.
15. Сайдашев А.А., Хеннер Е.К. Компьютер на уроке математики. – Пермь; Изд-во ПГУ, 1991.
16. Кондаков В.М. Математическое программирование. – Пермь: Изд-во ПГУ, 1992.
17. Матюшкин-Герке А. Учебно-прикладные задачи в курсе информатики. Информатика и образование, №3-4, 5-6, 1992.