**Лекция 4. Структурное моделирование**

1. Понятие структурной модели

Процессы и явления, происходящие в нашей Вселенной сложны и разнообразны: загадки человеческого разума, тайна рождения новой звезды во Вселенной, причины появления тайфунов, физика ядерных процессов, химические процессы горения, исчезновение видов животных, появление новых вирусов, тонкости человеческой психики и управление человеческими отношениями.

Наиболее часто встречается следующие виды моделирования:

* Моделирование физических процессов;
* Моделирование экологических систем;
* Моделирование математических объектов;
* Геоинформационное моделирование;
* Оптимизационное моделирование в экономике;
* Структурное моделирование.

Рассмотрим структурное моделирование. Важной характеристикой всякой системы является ее структура.

*Структура —* это определенный порядок объединения элементов, составляющих систему, иначе говоря, структура — это множество связей между элементами системы.

Наиболее удобным и наглядным способом представления структуры систем являются графы. Важной разновидностью графов являются деревья.

*Дерево* — это графическое представление иерархической структуры системы. Обычно это системы, между элементами которых установлены отношения подчиненности или вхождения друг в друга: системы власти, административные системы, системы классификации в природе и др. Ученики знакомы с понятием «дерево» применительно к системе файлов на дисках компьютера. Многим из них известен смысл понятия «родословное дерево».

*Структурное моделирование* – это создание и исследование модели, структура которой подобна структуре моделируемого объекта.

Варианты структурных моделей:

* графы;
* структурные и функциональные схемы;
* диаграммы;
* базы данных.
1. Структурные модели в среде СУБД

Для хранения и обработки больших объемов информации используются базы данных. Телефонный справочник является базой данных, в которой хранится информация об организациях (адрес, телефон и т. д.). Записная книжка является базой данных, в которую записывается информация о людях (фамилия, телефон, адрес электронной почты и т. д.). Библиотечный каталог является базой данных, которая хранит информацию о книгах (название, автор, год издания и т. д.).

Каждая база данных хранит информацию о большом количестве объектов одинакового типа (организациях, людях, книгах и т. д.). Объекты одного типа обладают одинаковым набором свойств, поэтому база данных хранит для каждого объекта значения этих свойств.

База данных позволяет упорядоченно хранить данные о большом количестве однотипных объектов, обладающих одинаковым набором свойств.

На этой структурной модели мы остановимся поподробнее.

Для того, чтобы управлять базой данных существует большое количество программ, иначе эти программы называют системами управления базами данных (СУБД). Создание баз данных, а также операции поиска и сортировки данных выполняются специальными программами – системами управления базами данных (СУБД). Таким образом, необходимо различать собственно базы данных, которые являются упорядоченными наборами данных, и системы управления базами данных – приложения, управляющие хранением и обработкой данных.

Этапы моделирования в СУБД:

Этап 1. Постановка задачи – упорядочение данных по некоторым признакам.

Этап 2. Разработка модели:

* информационная модель – выделение группы исходных данных;
* компьютерная модель – формирование структуры базы данных и ее заполнение данными.

Этап 3. Компьютерный эксперимент – манипулирование данными в СУБД.

Этап 4. Анализ результата – выводы, сделанные на основе анализ информации, представленные в отсчётах.

Общая схема этапов решения практической задачи на ЭВМ методами информационного моделирования выглядит следующим образом:

Два первых этапа относятся к предметной области решаемой задачи. На третьем этапе происходит выбор подходящего инструментального средства в составе программного обеспечения ЭВМ для реализации модели. Такими средствами могут быть: электронные таблицы, СУБД, системы программирования, математические пакеты, специализированные системы моделирования общего назначения или ориентированные на данную предметную область. В базовом курсе информатики изучаются первые три из перечисленных программных средств.

Основные признаки компьютерной информационной модели:

* наличие реального объекта моделирования;
* отражение ограниченного множества свойств объекта по принципу целесообразности;
* реализация модели с помощью определенных компьютерных средств;
* возможность манипулирования моделью, активного ее использования.

Ответ на вопрос: «является ли база данных информационной структурной моделью?» будем искать, исходя их сформулированных ниже критериев.

Первый критерий: наличие предметной области, некоторого реального объекта (системы), к которым относится БД, практически всегда выполняется. Например, если в БД содержатся сведения о книгах в библиотеке, значит объектом моделирования является книжный фонд библиотеки.

Если БД содержит анкетные данные сотрудников предприятия, значит она моделирует кадровый состав этого предприятия. Если в БД хранятся сведения о результатах сдачи экзаменов абитуриентами в институт, следовательно, она моделирует процесс вступительных экзаменов и т.п.

Второй критерий**:** удовлетворение второму критериютакже несложно обосновать. Каждый из моделируемых объектов (как перечисленные выше, так и любые другие) обладает гораздо большим числом свойств, характеристик, атрибутов, чем те, что отражены в БД. Отбор атрибутов, включаемых в БД, происходит в процессе проектирования базы, когда главным критерием является критерий целесообразности, т. е. соответствия цели создания БД, требованиям к ее последующим эксплуатационным свойствам. Например, в БД книжного фонда библиотеки не имеет смысла вносить такие характеристики книги, как ее вес, адрес типографии, где была напечатана книга, годы жизни автора и пр.

Третий критерий:очевидно, выполняется, поскольку речь идет о компьютерной базе данных, созданной в среде некоторой СУБД.

База данных — не «мертвое хранилище» информации. Она создается для постоянного, активного использования хранящейся в ней информации. Прикладные программы или СУБД, обслуживающие базу данных, позволяют ее пополнять, изменять, осуществлять поиск информации, сортировку, группировку данных, получение отчетных документов и пр. Таким образом, четвертый критерийкомпьютерной информационной структурной модели также справедлив для БД.

Проектирование базы данных заключается в теоретическом построении информационной модели определенной структуры.

Известны три основные структуры, используемые при организации данных в БД:

* иерархическая (деревья);
* сетевая;
* табличная (реляционная).

Доказано, что табличная структура является универсальной и может быть применена в любом случае.

Если объект моделирования представляет собой достаточно сложную систему, то проектирование БД становится нетривиальной задачей. Для небольших учебных БД ошибки при проектировании не столь существенны. Но если создается большая база, в которой будут сохраняться многие тысячи записей, то ошибки при проектировании могут стоить очень дорого. Основные последствия неправильного проектирования — избыточность информации, ее противоречивость, потеря целостности, т.е. взаимосвязи между данными. В результате БД может оказаться неработоспособной и потребовать дорогостоящей переделки.

Теория реляционных баз данных была разработана в 1970-х гг. Е. Коддом. Он предложил технологию проектирования баз данных, в результате применения которой в полученной БД не возникает отмеченных выше недостатков. Сущность этой технологии сводится к приведению таблиц, составляющих БД, к третьей нормальной форме.

Этот процесс называется нормализацией данных,сначала все данные, которые планируется включить в БД, представляются в первой нормальной форме, затем преобразуются ко второй и на последнем шаге — к третьей нормальной форме.

Пример 1. База данных «Словарь английских терминов по информатике».

Шаг 1. Создание таблиц

Для решения поставленной задачи принята схема базы данных, представленная на рисунке 1. Для открытия схемы данных необходимо выбрать команду «Схема данных» в меню «Сервис». В данной схеме представлена взаимосвязь всех таблиц базы данных и отношения между ними.



Рисунок 1 – Схема данных

Составляющими базу данных являются 2 таблицы: 1) перевод – включает в себя название термина и его толкование на английском языке; 2) толкование - включает в себя название термина и его толкование на русском языке.

Для создания таблицы в режиме конструктора необходимо:

1. открыть базу данных;
2. на вкладке «Таблицы» выбрать «Создание таблицы в режиме конструктора», откроется окно конструктора;
3. ввести имена полей таблицы и выбрать тип данных для каждого поля;
4. изменить свойства полей, если это необходимо;
5. выбрать ключевое поле, если это необходимо;
6. выполнить «Файл», «Сохранить» и сохранить таблицу, при этом если ключевое поле не было выбрано, Access предложит его создать;
7. закрыть таблицу.

Ниже описана структура основных таблиц «Перевод» и «Толкование».

Таблица «Перевод» предназначена для толкования термина на английском языке. Структура таблицы представлена в таблице 1.

Таблица 1 – Структура таблицы «Перевод»

|  |  |  |
| --- | --- | --- |
| ТИП ДАННЫХ | ПРИМЕНЕНИЕ | ПОЛЕ |
| Счетчик | Специальное числовое поле, в котором Access автоматически присваивает уникальный порядковый номер каждой записи. Значения полей типа счетчика обновлять нельзя. | Номер |
| Числовой | Числовые данные различных форматов, используемые для проведения расчетов. | Номер  |
| Текстовый | Текст или числа, не требующие проведения расчетов, например номера телефонов (до 255 знаков). | Термин, толкование |
| Дата/время | Для хранения календарных дат и текущего времени. | Дата |
| Денежный | Для хранения денежных сумм. | Цена |
| Поле объекта OLE | Объект (например, электронная таблица Microsoft Excel, документ Microsoft Word, рисунок, звукозапись или другие данные в двоичном формате), связанный или внедренный в таблицу Access. | Изображение |

Вид таблицы «Перевод» представлен на рисунке 2.



Рисунок 2 – Вид таблицы «Перевод»

Данная таблица в режиме конструктора выглядит следующим образом:



Рисунок 3 – Таблица «Перевод» в режиме конструктора

Таблица «Толкование» предназначена для толкования на русском языке. Структура таблицы представлена в таблице 2.

Таблица 2 – Структура таблицы «Толкование»

|  |  |  |
| --- | --- | --- |
| ТИП ДАННЫХ | ПРИМЕНЕНИЕ | ПОЛЕ |
| Счетчик | Специальное числовое поле, в котором Access автоматически присваивает уникальный порядковый номер каждой записи. Значения полей типа счетчика обновлять нельзя. | Номер  |
| Числовой | Числовые данные различных форматов, используемые для проведения расчетов. | Номер  |
| Дата/время | Для хранения календарных дат и текущего времени. | Дата |
| Денежный | Для хранения денежных сумм. | Сумма |

Вид таблицы «Толкование» представлен на рисунке 4.



Рисунок 4 – Вид таблицы «Толкование»

Режим конструктора позволяет изменять свойства полей, их название, определяет ключевые поля, типы данных, также позволяет оставлять комментарии, именно это и отображает рисунок 5. Последнее поле необязательно для заполнения.



Рисунок 5 – Режим конструктора

Вся необходимая работа осуществляется в специальном окне «Схема данных», которое открывается кнопкой  на панели инструментов или командой «Сервис» → «Схема данных».

Одновременно с открытием окна «Схема данных» открывается диалоговое окно «Добавление таблицы», из которого выбираются таблицы, между которыми создаются связи.

Добавление таблиц осуществляется выделением их в списке и щелчком на кнопке «Добавить». После добавления всех необходимых таблиц, окно «Добавление таблиц», закрывается щелчком на кнопке «Закрыть».

В итоге в окне «Схема данных» будут представлены все таблицы, создаваемой базы данных со списками своих полей. Ключевые поля выделены жирным шрифтом.

Шаг 2. Построение запросов в режиме «Конструктора»

В объекте «Запросы» есть специальный значок «Создание запроса» в режиме конструктора, который открывает специальный бланк, называемый бланком запроса по образцу. В верхней области отображается структура таблиц, к которым запрос адресован, а нижняя область разбита на столбцы – по одному на каждое поле будущей результирующей таблицы. При создании запроса на основе нескольких взаимосвязанных таблиц в бланк запроса необходимо поместить все участвующие в запросе таблицы.

Таблицы, к которым обращен запрос, можно открыть с помощью контекстного меню. Далее, с помощью двойного щелчка мышью на нужном поле, его можно поместить в столбец нижней части бланка.

Если необходимо, чтобы отобранные в результате выполнения запроса данные, были упорядочены по какому-либо полю, применяют «Сортировку», которую можно увидеть в нижней части бланка. Возможна многоуровневая сортировка – сразу по нескольким полям. Вложенность сортировки идет слева направо.

Бывают случаи, когда некоторое поле необходимо в формировании запроса, но нежелательно, чтобы оно появлялось на экране. В таких случаях сбрасывают флажок для этого поля в строке «Вывод на экран» в нижней части бланка.

Дополнительным средством, обеспечивающим отбор данных по заданному критерию, является «Условие отбора». Соответствующая строка также присутствует в нижней части бланка запроса. Для каждого поля можно задать индивидуальное условие, используя логические отношения, причем, заполнение одной строки для разных полей, означает, что они соединены логической операцией AND. Если для одного поля логические отношения расположены в двух строках, они соединены логической операцией OR.

Для изменения следования полей в столбцах нижней части бланка необходимо:

1. выделить столбец щелчком на его заголовке (кнопку мыши отпустить);
2. еще раз щелкнуть на заголовке и, не отпуская кнопки, перетащить столбец на нужное место.

Для выполнения запроса нужно нажать кнопку  на панели инструментов или перейти в режим таблицы, выполнив команду «Вид» → «Режим таблицы».

При закрытии для сохранения запроса ему дают имя.

Далее создаются запросы. **Запрос** – это средство выбора необходимой информации из базы данных. Вопрос, сформированный по отношению к базе данных, и есть запрос.



Рисунок 6 – Запрос 1: поиск по терминам на русском языке

Вводим в окно название термина, получаем его толкование. Окно ввода термина представлено на рисунке 6, а окно вывода результата – на рисунке 7.



Рисунок 7 – Толкование на русском языке

Тоже самое делается и на английском языке.

Запросы в БД создаются с помощью конструктора запросов.



Рисунок 8 – Конструктор запросов

В этом режиме можно редактировать, изменять свойства, скрывать или выводить на экран поля запросов.

Шаг 3. Создание форм

Для создания автоформы следует открыть панель «Формы» в окне «База данных» и воспользоваться командной кнопкой . В открывшемся диалоговом окне «Новая форма» выбирают тип автоформы и таблицу, на которой она основывается. После щелчка на кнопку «ОК» автоформа формируется автоматически и немедленно готова к работе, то есть к вводу или отображению данных.

Автоформа основывается только на одном объекте, она называется еще простой формой. Если форма основывается на полях из нескольких связанных таблиц, то она называется сложной. Для ее создания используют другие средства.

Уточнение текста надписей, местоположения, размеров, шрифтов и других параметров отображения элементов формы осуществляется в режиме «Конструктора форм».

Для открытия формы в режиме конструктора следует выделить необходимую форму на панели «Формы» в окне «База данных» и воспользоваться командной кнопкой .

Если форма уже была открыта, перейти в режим конструктора можно, используя командную кнопку «Вид».

При запуске базы данных автоматически запускается «Главная кнопочная форма», созданная средствами ACCESS. Она представляет собой удобный интерфейс для работы и навигации по БД. Для её создания необходимо выполнить «Сервис» → «Служебные программы» → «Диспетчер кнопочных форм». Откроется окно диспетчера кнопочных форм. Далее создаём необходимые подчиненные кнопочные формы, а затем изменяем их, наполнив кнопками. При этом необходимо для каждой кнопки указать название, выполняемое ею действие и если необходимо – объект, над которым выполняется это действие.

Также в этом словаре используются формы. Форма — это объект, с помощью которого пользователи могут добавлять, редактировать и отображать данные, хранящиеся в базе данных. В данной БД можно посмотреть все термины подряд на русском и английском языках, используя кнопки навигации, как представлено на рисунке 9.



Рисунок 9 – Форма: словарь в алфавитном порядке

Ну и наконец главное назначение данного словаря – это перевод терминов с одного языка на другой.



Рисунок 10 – Форма: толкование\_перевод

Данные формы представимы в режиме конструктора, как на рисунке 11.



Рисунок 11 – Формы: режим конструктора

Режим конструктора форм предназначен для создания и редактирования форм.

Основным свойством таблиц является их связанность. Использование этого свойства позволяет избежать избыточность, переполненность информацией.

В данной БД используется тип связи «один к одному», это видно по рисунку 12. Данный вид связи используется для того, чтобы избежать ошибок. Термину «Интернет» строго должно соответствовать его толкование, его перевод и т.д.

Рисунок 12 - Связи

1. Структура обучающего сценария в MC Excel

Среда Microsoft Excel интересна многочисленными средствами автоматизации работы, оформления документов и богатыми вычислительными возможностями. Загадочность ее состоит в том, что большинство пользователей применяют лишь малую толику того, что может дать им Excel. Это тем более удивительно, что спектр возможностей программы практически безграничен: от создания простых таблиц, построения диаграмм и графиков до решения сложных вычислительных задач и моделирования различных процессов.

С середины XX века в самых различных областях человеческой деятельности стали широко применять математические методы и ЭВМ. Возникли такие новые дисциплины, как «математическая экономика», «математическая химия», «математическая лингвистика» и т.д., изучающие математические модели соответствующих объектов и явлений, а также методы исследования этих моделей.

Процесс разработки моделей и их исследование в табличном процессоре можно разделить на несколько основных этапов.

Этап 1. Описательная информационная модель. Такая модель выделяет существенные, с точки зрения целей проводимого исследования, параметры объекта, а несущественными параметрами пренебрегает.

Этап 2. Формализованная модель*,* т.е. описательная информационная модель записывается с помощью какого-либо формального языка. В такой модели с помощью формул, уравнений, неравенств и т.д. фиксируются формальные соотношения между начальными и конечными значениями свойств объектов, а также накладываются ограничения на допустимые значения этих свойств.

Этап 3. Компьютерную модель, т.е. выразить ее на понятном для компьютера языке с использованием одного табличного процессора Excel.

Этап 4. Компьютерный эксперимент. После создания компьютерной модели в Excel, ее нужно запустить на выполнение и получить результаты. Если компьютерная модель исследуется в приложении, то можно провести сортировку или поиск данных, построить диаграмму или график и т.д.

Этап 5. Анализ полученных результатов и корректировка исследуемой модели. В случае различия результатов, полученных при исследовании информационной модели, с измеряемыми параметрами реальных объектов, можно сделать вывод, что на предыдущих этапах построения модели были допущены ошибки или неточности.

Пример 1. Выбор профессий из последовательности слов на английском языке.

Полный код программы выглядит так:

Таблица 3 – Код программы «Профессии»

|  |  |
| --- | --- |
| Public count As Integer, isStart As Boolean | Объявление переменных |
| *'Обработка для кнопки Старт* |  |
| Private Sub Start\_Click() | Процедура кнопки Старт |
| Dimw() AsVariant | Объявление переменных |
| Dim lGame, lSett As Worksheet |
| Dim x, y As Integer |
| Dim mas(1 To 12, 1 To 11) As String |
|  |  |
| Set lGame = ThisWorkbook.Worksheets("Game") | Готовим поле с вопросом |
| Set lSett = ThisWorkbook.Worksheets("Setting") |
| count = 0 |
| Game.Range("Status") = "" |
|  |  |
| WithGame | Очистка |
| .Range("GamePlace").ClearContents |
| .Label1.Caption = lSett.Range("Artic") |
| .result.Caption = "" |
|  |  |
| isStart = True | Подготовка переменных |
|  |  |
| DoEvents | Реализуем события рабочего стола |

Продолжение таблицы 3

|  |  |
| --- | --- |
| ReDim w(1 To lSett.Range("words").count, 1 To 1) |  |
| w = lSett.Range("words").Value | Помещаем значение в переменную |
| For Each d In w | Цикл |
| nxt: | Указатель ГОТО |
| x = Int(Rnd \* 10) + 1 | Выбираем случайную ячейку |
| y = Int(Rnd \* 10) + 1 |
|  |  |
| If mas(y, x) = Empty Then | Если она пуста |
| mas(y, x) = d | Присваиваем ей знаение d |
| Else | Иначе |
| GoTo nxt | Безусловный переход к указателю |
| End If | Конец условия |
|  |  |
| Next d | Переход к очередному слову |
| Game.Range("GamePlace") = mas |  |
| End With | Нижняя граница цикла |
| End Sub | Конец процедуры Start\_Click() |
|  |  |
| *'Выбор слов (выбор ячеек)* |  |
| Private Sub Worksheet\_SelectionChange(ByVal Target As Range) | Процедура выбора слов |
| Dim kol As Integer | Объявление переменных |

Продолжение таблицы 3

|  |  |
| --- | --- |
| Dim lGame, lSett As Worksheet |  |
| Set lGame = ThisWorkbook.Worksheets("Game") |  |
| Set lSett = ThisWorkbook.Worksheets("Setting") |  |
| kol = lSett.Cells(1, 4) | Получаем в переменную данные |
|  |  |
| If Target.count = 1 And Target.Row> 6 And Target.Row< 19 And Target.Column< 15 And Target.Row> 4 Then | Проверяем попадание в допустимый диапазон |
| If isStart And Not Target = Empty Then | Проверяем условие |
| Set rr = lSett.Range("words").Find(Target.Value) | Получаем в переменную данные |
| If Not rr Is Nothing Then | Проверяем условие |
| f lSett.Range("word\_stat")(rr.Row) = 1 Then | Проверяем условие |
| count = count + 1 | Увеличиваем переменную счетчик на 1 |
| lGame.result.Caption = lGame.result.Caption& " " &Target.Value | Выводим данные |
| lGame.Range("Status") = "Верно! Найдите оставшиеся профессии!" | Выводим сообщение |
| lGame.Range("Status").Font.Color = -14792145 | Меняем цвет |
| lGame.Cells(Target.Row, Target.Column).ClearContents | Очищаем  |

Продолжение таблицы 3

|  |  |
| --- | --- |
| lGame.Range("Status").Select | Выбираем область |
| If count = kol Then |  |
| isStart = False | Устанавливаем в ложь |
| lGame.Range("GamePlace").ClearContents | Очищаем |
| lGame.Range("Status") = "Поздравляю! Вы нашли все профессии!" | Выводим сообщение |
| lGame.Range("Status").Font.Color = -14792145 | Меняем цвет |
| lGame.Label1.Caption = lSett.Range("start") | Выводим данные |
| End If | Конец условия |
| Else | Иначе |
| lGame.Range("Status") = "Неправильно! Попробуйте еще!" | Выводим сообщение |
| lGame.Range("Status").Font.Color = vbRed | Меняем цвет на красный |
| End If | Конец условия |
| End If |
| End If |

Пользователю предлагается выбрать среди данных английских слов только те, которые обозначают профессии.



Рисунок 14 – Выбор профессии

При щелчке по «правильному» слову, оно попадает в окошко.



Рисунок 15 – Выбор правильного слова

Если выбрали «неправильное» слово, то программа оповещает пользователя об этом.



Рисунок 16 – Выбор неправильного слова

В случае правильного выбора очередное слово добавляется к последовательности слов и производится проверка. Если все сделано правильно, то поле очищается.



Рисунок 17 – Проверка выполненного задания

Пример 2. Составление предложения на английском языке.

Полный код программы выглядит так:

Таблица 4 – Код программы «Составление предложения»

|  |  |
| --- | --- |
| Public count As Integer, isStart As Boolean | Объявление переменных |
|  |  |
| *'Обработка для кнопки старт* |  |
| PrivateSubStart\_Click() |  |
|  |  |
| Dim w() As Variant | Объявление переменных |
| Dim lGame, lSett As Worksheet |
| Dim x, y As Integer |
| Dim mas(1 To 12, 1 To 11) As String |  |
|  |  |
| Set lGame = ThisWorkbook.Worksheets("Game") | Создание рабочего пространства |
| Set lSett = ThisWorkbook.Worksheets("Setting") |
| count = 0 |
| Game.Range("Status") = "" |
|  |  |
| With Game |  |
| .Range("GamePlace").ClearContents | Очищаем области данных |
| .Label1.Caption = lSett.Range("Artic") | Готовим элемент управления для ввода предложения |
| .result.Caption = "" |
|  |  |
| isStart = True | Подготовка переменных |
| Set lr = lSett.Range("words")(1) |

Продолжение таблицы 4

|  |  |
| --- | --- |
| Do Events | Реализуем события рабочего стола |
| ReDim w(1 To lSett.Range("words").count, 1 To 1) | Начинаем цикл для заполнения рабочего листа словами |
| w = lSett.Range("words").Value |
| For Each d In w |
| nxt: | Указатель ГОТО |
|  x = Int(Rnd \* 10) + 1 | Выбираем случайную ячейку |
|  y = Int(Rnd \* 10) + 1 |
| If mas(y, x) = Empty Then | Если она пуста |
| mas(y, x) = d | Присваиваем ячейке текст из d |
|  Else | Иначе |
| GoTo nxt | Безусловный переход выше, к указателю nxt  |
| End If | Конец условия |
|  |  |
| Next d | Переход к очередному слову |
| Game.Range("GamePlace") = mas |  |
| End With | Конец цикла |
| End Sub | Конец процедуры |
|  |  |
| *'Выбор слов (выбор ячеек)* |  |
| Private Sub Worksheet\_SelectionChange(ByVal Target As Range) | Процедура выбора ячеек |
|  |  |
| Dim lGame, lSett As Worksheet  | Объявление переменных |

Продолжение таблицы 4

|  |  |
| --- | --- |
| Dim kol As Integer |  |
| Set lGame = ThisWorkbook.Worksheets("Game") |  |
| Set lSett = ThisWorkbook.Worksheets("Setting") |  |
| kol = lSett.Cells(1, 4) |  |
| If Target.count = 1 And Target.Row> 6 And Target.Row< 19 And Target.Column< 15 And Target.Row> 4 Then | Должна быть выделена только одна и не пустая ячейка |
| If isStart And Not Target = Empty Then |  |
| If lSett.Range("words")(count + 1) = Target Then | Если номер выбранного тот, который нам нужен, то |
| count = count + 1 | Счетчик увеличиваем на один |
| lGame.result.Caption = lGame.result.Caption& " " &Target.Value | Добавляем угаданное слово |
| lGame.Range("Status") = "Верно! Подставьте оставшиеся слова!" | Выводим надпись |
| lGame.Range("Status").Font.Color = -14792145 | Меняем цвет |
| lGame.Cells(Target.Row, Target.Column).ClearContents | Очищаем |
| lGame.Range("Status").Select | Выбираем область |
| If count = kol Then |  |
| isStart = False | Если «да», то isStart = False |
| lGame.Range("GamePlace").ClearContents | А все лишние слова удаляются |

Продолжение таблицы 4

|  |  |
| --- | --- |
| lGame.Range("Status") = "Поздравляю! Вы составили предложение правильно!" | Выводим надпись |
| lGame.Range("Status").Font.Color = -14792145 | Меняем цвет |
| lGame.Label1.Caption = lSett.Range("start") | Выводим надпись |
| End If | Конец условия |
| Else | Иначе |
| lGame.Range("Status") = "Неправильно! Попробуйте еще!" | Выводим сообщение |
| lGame.Range("Status").Font.Color = vbRed | Меняем цвет на красный |
| End If | Конец условия |
| End If |
| End If |
| End Sub | Конец процедуры SelectionChange |

Пользователю предлагается составить предложение на английском языке.



Рисунок 18 – Составьте предложение

При щелчке по «правильному» слову, оно попадает в окошко.



Рисунок 19 – Выбор первого слова

Если выбрали «неправильное» слово, то программа оповещает пользователя об этом.



Рисунок 20 – Выбор неправильного слова

В случае правильного выбора очередное слово добавляется к предложению и производится проверка на длину предложения. Если предложение правильное, то текст заканчивается.



Рисунок 21 – проверка предложения

1. Задание для самостоятельной работы

Разработать следующие тесты:

1. Ответ на загадку;
2. Выбор животных из последовательности слов;
3. Составить предложение из представленных слов.