**СОДЕРЖАНИЕ**

Введение…………………………………………………………………… 3

Глава 1. Теоретическая часть…………………………………………….. 4

Глава 2. Практическая часть.……………………………………………... 8

Заключение…………………….…….…………………………………... 12

Список используемой литературы……………………………………… 13

Приложение 1. Руководство пользователя…………………………….. 14

Приложение 2. Листинг программы……………………………………. 20

## Введение

В XXI веке в развитии человеческой цивилизации происходят глобальные изменения, ведущие к её новому этапу - постиндустриальному обществу, все шире использующему компьютеризированные орудия труда и информационные технологии.

Стремительное развитие индустрии, влечет за собой все больший и больший интерес начинающих программистов создавать то, что актуально в наше время. Поэтому, для выполнения курсовой работы мы выбрали учет успеваемости студентов. Этот процесс достаточно трудоемкий и может занимать много времени. Мы в своей курсовой работе предлагаем упрощенный и автоматизированный вариант данного учета.

Данная программа, написанная на языке программирования Visual Basic for Application, представляет собой учет успеваемости студентов.

Курсовая работа проста, удобна и практична в использовании.

Курсовая работа содержит: введение, теоретическую часть, практическую реализацию, заключение, литературу, которая использовалась при написании курсовой работы и приложения.

## Глава 1 .Теоретическая часть

## 1.1 Постановка задачи

Задача данного курсового проекта - разработка информационной системы учета успеваемости студентов, а также ее интерфейса.

## 1.2 Анализ задачи

В данной курсовой работе разработана информационная система учета успеваемости студентов. Эта тема является актуальной в настоящее время, так как многие учебные заведения проводят учет успеваемости учащихся для отслеживания уровня подготовки и выявления возможных недостатков в образовательном процессе.

Система учета успеваемости студентов должна обеспечивать:

1. Поиск и сортировку информации:

- по каждому студенту;

- по группам;

- по факультетам;

- по каждой дисциплине.

2. Добавление данных о студентах.

3. Удаление данных.

4. Редактирование имеющихся данных.

5. Формирование карточек и списков.

6. Построение диаграмм.

На практике это означает выполнение автоматизированной обработки анализа успеваемости студентов; освобождение сотрудников от ряда рутинных работ по подготовке отчетной документации; создание базы данных о студентах; осуществление операций по созданию архивов успеваемости студентов. Благодаря автоматизации можно выполнять поиск информации по запросам преподавателей.

## 1.3. Обоснование и выбор инструментальных средств

Практическая реализация курсового проекта осуществлялась в операционной системе Windows, с использованием электронной таблицы Excel из пакета Microsoft Offiсe.

Табличный процессор Excel является одним из наиболее популярных пакетов программ, предназначенных для создания табличных документов. Предлагаемые им средства позволяют решать разнообразные задачи от расчета величины подоходного налога до составления финансового отчета крупной компании. Система обладает мощными вычислительными возможностями, великолепными средствами составления деловой графики, обработки текстов, введения баз данных. Работа с табличным процессором Excel позволяет:

- использовать для хранения взаимосвязанных таблиц рабочую книгу, состоящую из отдельных листов, которые можно в процессе работы удалять, переименовывать, переставлять местами, копировать и скрывать;

- применять удобные средства корректировки данных в таблице, использовать широкий спектр возможностей работы с фрагментами;

- осуществлять должное оформление таблиц - применять разнообразные шрифты, способы выравнивания текста и чисел, изменения ширины столбцов и высоты строк, затемнения и обрамления ячеек;

- применять разнообразные форматы отображения числовых данных;

- для графического представления данных рабочего листа применять широкий набор диаграмм, должным образом оформлять их и печатать;

- выполнять свод данных из нескольких таблиц путем их консолидации.

Пользовательский интерфейс приложения разрабатывался на базе языка Visual Basic for Application.

В процессе создания собственных прикладных программ на языке VBA возникает необходимость реализации собственных диалоговых окон. Для этого в VBA есть мощное инструментальное средство – конструктор диалогов.

Большинство программ в системе Windows используют выпадающее меню, облегчающее доступ к директивам приложения и управляющее работой этого приложения. Более современный подход - использование пиктографического меню, обеспечивающего более простой доступ к основным директивам программы. Visual Basic позволяет создавать собственные пользовательские и пиктографические меню. VBA дает возможность записывать макросы с помощью макрорекордера, а также просматривать листинг записанного макроса.

VBA также содержит средства отладки программы – Debug, что значительно упрощает корректировку и отладку модулей на этапе проектирования приложения.

Режим просмотра объектов с помощью редактора VBA позволяет:

- раскрыть список доступных объектов;

- определить принадлежность того или иного объекта к библиотеке объектов;

- осуществить просмотр свойств и методов для выбранного объекта;

- получить справку по конкретному объекту, его свойству или методу.

Язык программирования Visual Basic for Application дает возможность проектировать приложения с простым и надежным пользовательским интерфейсом.

## 1.4. Виды диалоговых окон и основных элементов управления

Для создания диалоговых окон используется конструктор диалогов, позволяющих создавать диалоговые окна пользователя. Для того чтобы загрузить конструктор диалогов, необходимо выполнить команду Вставка/UserForm редактора VBA. После этого на экране появится окно диалога и пиктографическое меню Панель элементов.

Каждая пиктограмма меню Панель элементов дает возможность изобразить на диалоге различные элементы управления, позволяющие скомпоновать диалоговое окно пользователя. Каждый элемент управления является объектом, поэтому обладает рядом свойств и методов.

Все элементы управления диалогового окна пользователя объединяются набором Controls. Используя данный набор, можно получить доступ к любому элементу управления на форме, к его свойствам и методам. Применение данного набора удобно в тех случаях, когда последовательно в цикле перебираются элементы формы, определяется их количество, задаются значения тех или иных общих свойств. Список свойств выбранного элемента управления, отображается в окне свойств редактора VBA. Таким образом, свойства элемента управления можно менять как вручную, так и программным путём.

Элементы управления делятся на стандартные и дополнительные. Рассмотрим стандартные элементы управления.

К стандартным элементам управления относятся кнопки, списки, окна редактирования, линейки прокрутки и т.п.

В моей курсовой работе использовались следующие элементы управления:

● Окно редактирования (имя, фамилия, отчество, группа, список предметов) позволяют вводить, изменять и просматривать данные.

● Линейка прокрутки позволяет просматривать оценки студентов за семестр по конкретному предмету.

● Кнопки для просмотра имеющихся записей, перехода от одной записи к другой, добавления и удаления записей и кнопка для выхода из программы.

## Глава 2. Практическая часть

После анализа структуры входящих потоков информации, для данного проекта был выбран интерфейс, соответствующий основным требованиям и обеспечивающий легкость работы с ним. Интерфейс – это совокупность средств, обеспечивающих физическое или логическое взаимодействие устройств и программ вычислительной системы. Для данного проекта интерфейс – изобретение и создание такого способа ввода и вывода информации, который был бы максимально прост, удобен и приятен пользователю.

## 2.1 Описание интерфейса

# Окно 1 «курсовая работа»

Стартовое окно программы. Это окно инициализируется при открытии документа. В форму добавлены фоновый рисунок и пять элементов управления:

* Элемент «Редактирование базы данных» (CommandButton1) – с событием Click была связана процедура Private Sub Edit\_Click(), при выполнении которой осуществляется открытие соответствующего окна.
* Элемент «Создание отчета» (CommandButton2) - с событием Click была связана процедура Private Sub Report\_Click(), при выполнении которой появляется окно «Отчеты».
* Элемент «Диаграмма успеваемости группы» (CommandButton3) - с событием Click была связана процедура Private Sub DiaGroup\_Click(). При ее выполнении осуществляется открытие окна с аналогичным названием.
* Элемент «Диаграмма успеваемости студента» (CommandButton4) - с событием Click была связана процедура Private Sub DiaStudent\_Click(). При выполнении открывается окно с аналогичным названием.
* Элемент «Выход» (CommandButton5) - с событием Click была связана процедура Private Sub Quit\_Click() – выход из программы.

**Окно 2 «Отчет»**

Данное окно используется для создания отчета по успеваемости студента или группы студентов. Форма содержит 3 элемента Label, 2 элемента ComboBox, 7 элементов CommandButton и 1 элемент ListBox.

● Кнопка «Создать» (CommandButton1) – с событием Click связана процедура Private Sub Create\_Click(), при выполнении которой создается отчет в соответствии с заданными параметрами.

● Кнопка «Закрыть» (CommandButton2) – с событием Click связана процедура Private Sub Cancel\_Click() – выход из формы создания отчетов.

● Кнопка «Очистить» (CommandButton3) – с событием Click связана процедура Private Sub Clear\_Click(), при выполнении которой происходит очищение элемента ListBox.

● Кнопка «Вверх» (CommandButton4) – с событием Click связана процедура Private Sub Up\_Click(). С помощью этой кнопки можно менять местами студентов при создании отчетов.

● Кнопка «Вниз» (CommandButton5) – с событием Click связана процедура Private Sub Down\_Click(). С помощью этой кнопки можно менять местами студентов при создании отчетов.

● Кнопка «>>» (CommandButton6) – с событием Click связана процедура Private Sub Add\_Click(). С помощью этой кнопки можно добавлять выбранных студентов в ListBox для создания отчетов их успеваемости.

● Кнопка «<<» (CommandButton7) – с событием Click связана процедура Private Sub Remove\_Click(). С помощью этой кнопки можно удалять выбранных студентов из ListBox.

**Окно 3 «Редактирование базы данных»**

С помощью этого окна можно изменять список студентов, их успеваемость по конкретному предмету, добавлять новые записи, удалять, а также просматривать уже имеющуюся информацию. Форма содержит следующие элементы управления: 8 элементов Label, 6 элементов TextBox, 1 элемент ListBox, 2 элемента SpinButton, 7 элементов CommandButton.

● Кнопка «Добавить» (CommandButton1) с событием Click связана процедура Private Sub Update\_Click(), при выполнении которой добавляется новая запись.

● Кнопка «Удалить» (CommandButton2) с событием Click связана процедура Private Sub Delete\_Click(), при выполнении которой запись удаляется.

● Кнопка «Закрыть» (CommandButton3) с событием Click связана процедура Private Sub Cancel\_Click(), при выполнении которой закрывается окно.

● Кнопка «<<» (CommandButton4) с событием Click связана процедура Private Sub Begin\_Click(), которая переходит к первому элементу списка.

● Кнопка «>>» (CommandButton5) с событием Click связана процедура Private Sub NewItem\_Click(), которая очищает поля для создания новой записи.

● Кнопка «<» (CommandButton7) с событием Click связана процедура Private Sub Backward\_Click(), при выполнении которой происходит переход к предыдущему элементу списка.

**Окно 4 «Диаграмма успеваемости группы»**

С помощью этого окна мы можем построить диаграммы успеваемости группы по любому конкретному предмету. Форма содержит следующие элементы управления: 4 элемента Label, 2 элемента ComboBox, 2 элемента CommandButton, 2 элемента OptionButton.

● Кнопка «Создать» (CommandButton1) с событием Click связана процедура Private Sub Create\_Click(), которая создает диаграмму по заданным параметрам.

● Кнопка «Закрыть» (CommandButton2) с событием Click связана процедура Private Sub Cancel\_Click(), при выполнении которой закрывается окно.

**Окно 5 «Диаграмма успеваемости студента»**

С помощью этого окна мы можем построить диаграммы успеваемости студента. Форма содержит следующие элементы управления: 4 элемента Label, 2 элемента ComboBox, 2 элемента CommandButton, 2 элемента OptionButton.

● Кнопка «Создать» (CommandButton1) с событием Click связана процедура Private Sub Create\_Click(), которая создает диаграмму по заданным параметрам.

● Кнопка «Закрыть» (CommandButton2) с событием Click связана процедура Private Sub Cancel\_Click(), при выполнении которой закрывается окно.

**Модули:**

● Sub QDialog1() – показывает форму «Курсовая работа».

● Sub Macro1() – устанавливает ширину трех столбцов A, B, C.

● Sub Macro2() - Оформление списка отчета (с помощью записи макроса).

## Заключение

В данной курсовой работе была разработана программа учета успеваемости студентов.

Программа была разработана с помощью языка программирования Visual Basic for Application (VBA).

Язык программирования Visual Basic for Application – быстрое, простое решение многих задач, связанных с программированием офисных приложений.

На примере данной курсовой работы можно убедиться, как просто создаются диалоговые окна, для наиболее удобного ввода информации в таблицы как прост и удобен интерфейс данной системы. Также просто создаются собственные функции рабочего листа, функции обработки информации в таблицах, создавать кнопки управляющие действиями программы по желанию пользователя.

Visual Basic for Application обладает достаточно широкими возможностями базового языка Visual Basic, и позволяет обращаться практически ко всем возможностям Microsoft Excel по созданию, обработке и хранению информации.

Язык программирования Visual Basic for Application наиболее удобен в работе и прост в обращении для широкого круга пользователей, желающих самостоятельно научиться создавать программы.

Таким образом, данная программа по учету успеваемости студентов делает наиболее простой работу пользователя в данной области и сокращает время на выполнение соответствующих операций.

## Список используемой литературы

1. Комолова Н.И., Программирование на VBA в Excel, BHY, 2019.
2. Швыдков С.А., Учебник по функциям Excel и программированию в среде VBA,МГУ, 2019.
3. Макграт М., Excel VBA, Экстмо, 2020.
4. У. Орвис «Visual Basic for Application», БИНОМ, М., 1995.
5. Э. Бунин «Excel Visual Basic для приложений», БИНОМ, М., 1996.
6. Visual Basic Help Reference File, Microsoft Corporation, 1996-97.

Приложения

**ПРИЛОЖЕНИЕ 1**

**РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

При открытии файла появляется следующее диалоговое окно:

**Окно 1 «Курсовая работа»**

#

Рис.1

На нем расположены несколько кнопок: редактирование базы данных, создание отчета, диаграмма успеваемости группы, диаграмма успеваемости студента, выход. А также сведения об авторах.

При нажатии кнопки «Редактирование базы данных» появляется диалоговое окно:

**Окно 2 «Редактирование базы данных»**



Рис.2

С помощью этого окна можно изменять список студентов, их успеваемость по конкретному предмету, добавлять новые записи, удалять, а также просматривать уже имеющуюся информацию.

При нажатии на кнопку «<<», мы переходим в начало списка.

При нажатии на кнопку «<», мы переходим к предыдущему элементу списка.

При нажатии на кнопку «>», мы переходим к следующему элементу списка.

При нажатии на кнопку «>>», мы очищаем поля для создания новой записи.

При нажатии на кнопку «Добавить», происходит добавление новой записи.

При нажатии на кнопку «Удалить», происходит удаление выделенной записи.

При нажатии на кнопку «Закрыть», происходит закрытие данного окна.

Из списка предметов с помощью мышки, мы выбираем предмет, с помощью линейки прокрутки можно изменять оценки студентов.

При нажатии кнопки «Создание отчета», появляется следующее диалоговое окно:

**Окно 3 «Отчет»**



Рис.3

Данное окно используется для создания отчета по успеваемости студента или группы студентов.

Кнопки «>>» и «<<» добавляют и удаляют студентов из списка для создания отчёта:



Рис.4

Кнопка «Очистить» - очищает поле списка студентов.

Кнопки «Вверх» и «Вниз» - позволяют перемещать студентов по списку.

При нажатии на кнопку «Создать» - появляется отчет по введенным параметрам:



Рис.5

При нажатии кнопки «Закрыть» - данное окно закрывается.

При нажатии на кнопку «Диаграмма успеваемости группы» появляется диалоговое окно:

**Окно 4 «Диаграмма успеваемости группы»:**



Рис.6

В поле «Группа» выбираем нужную из списка, в поле «Предмет» выбираем предмет и выбираем курсором 1-ый или 2-ой семестр. Нажимаем на кнопку «Создать». Появляется окно:



Рис.7

Кнопка «Закрыть» - для выхода.

При нажатии на кнопку «Диаграмма успеваемости студента» появляется окно:

**Окно 5 «Диаграмма успеваемости студента»:**



Рис.8

Выбираем группу, студента и семестр и создаем диаграмму по заданным параметрам:



Рис.9

Кнопка «Закрыть» - для выхода из окна.

При нажатии кнопки «Выход» в окне «Курсовая работа» осуществляется выход из программы.

**ПРИЛОЖЕНИЕ 2**

**ИСХОДНЫЙ КОД ПРОГРАММЫ**

**Форма «MainForm»:**

Private Sub DiaGroup\_Click()

DiaGroupForm.Show

End Sub

Private Sub DiaStudent\_Click()

DiaStudentForm.Show

End Sub

Private Sub Edit\_Click()

DataForm.Show

End Sub

Private Sub Label2\_Click()

End Sub

Private Sub Quit\_Click()

Unload Me

End Sub

Private Sub Report\_Click()

ReportForm.Show

End Sub

**Форма «ReportForm»:**

Dim Cursor As Integer

Private Sub Add\_Click()

For i = 0 To Queue.ListCount - 1

If Queue.List(i) = Group.Value + " - " + Student.Value Then

MsgBox "Такой элемент уже есть в очереди!"

Exit Sub

End If

Next i

Queue.AddItem Group.Value + " - " + Student.Value

End Sub

Private Sub Cancel\_Click()

Me.Hide

End Sub

Private Sub Clear\_Click()

Queue.Clear

End Sub

Private Sub Create\_Click()

Dim StudentVal As String

Dim GroupVal As String

Dim i As Integer

Dim j As Integer

Dim Value As String

Dim Row As Integer

Row = 2

ActiveWorkbook.Worksheets.Add After:=Worksheets(Worksheets.Count)

With ActiveSheet

.Cells.Clear

.Columns(1).ColumnWidth = 26

.Columns(2).ColumnWidth = 20

.Columns(3).ColumnWidth = 20

For i = 0 To Queue.ListCount - 1

GroupVal = Split(Queue.List(i), " - ")(0)

StudentVal = Split(Queue.List(i), " - ")(1)

j = 2

Do

Value = Worksheets("Storage").Cells(j, 1).Value

If Len(Value) > 0 Then

If Worksheets("Storage").Cells(j, 1) = GroupVal And Worksheets("Storage").Cells(j, 2) + " " + Worksheets("Storage").Cells(j, 3) + " " + Worksheets("Storage").Cells(j, 4) = StudentVal Then

.Cells(Row, 1).Value = "Группа"

.Cells(Row, 2).Value = GroupVal

Range(.Cells(Row, 2), .Cells(Row, 3)).MergeCells = True

.Cells(Row + 1, 1).Value = "Студент"

.Cells(Row + 1, 2).Value = StudentVal

Range(.Cells(Row + 1, 2), .Cells(Row + 1, 3)).MergeCells = True

Range(.Cells(Row, 2), .Cells(Row + 1, 2)).Select

With Selection.Interior

.ColorIndex = 36

.Pattern = xlSolid

End With

.Cells(Row + 2, 1).Value = "Оценки по предметам"

.Cells(Row + 2, 2).Value = "1-ый семестр"

.Cells(Row + 2, 3).Value = "2-ой семестр"

Range(.Cells(Row + 2, 2), .Cells(Row + 2, 3)).Select

With Selection

.HorizontalAlignment = xlCenter

.Interior.ColorIndex = 37

.Interior.Pattern = xlSolid

End With

For k = 0 To 8

.Cells(Row + 3 + k, 1).Value = Worksheets("Storage").Cells(1, k + 5).Value

.Cells(Row + 3 + k, 2).Value = Split(Worksheets("Storage").Cells(j, k + 5).Value, ":")(0)

.Cells(Row + 3 + k, 3).Value = Split(Worksheets("Storage").Cells(j, k + 5).Value, ":")(1)

Next k

Range(.Cells(Row + 3, 2), .Cells(Row + 2 + k, 3)).Select

With Selection

.HorizontalAlignment = xlCenter

.Interior.ColorIndex = 35

.Interior.Pattern = xlSolid

End With

Range(.Cells(Row, 1), .Cells(Row + 2 + k, 1)).Select

With Selection.Interior

.ColorIndex = 40

.Pattern = xlSolid

End With

Range(.Cells(Row, 1), .Cells(Row + 2 + k, 3)).Select

Selection.Borders(xlDiagonalDown).LineStyle = xlNone

Selection.Borders(xlDiagonalUp).LineStyle = xlNone

With Selection.Borders(xlEdgeLeft)

.LineStyle = xlContinuous

.Weight = xlThin

.ColorIndex = xlAutomatic

End With

With Selection.Borders(xlEdgeTop)

.LineStyle = xlContinuous

.Weight = xlThin

.ColorIndex = xlAutomatic

End With

With Selection.Borders(xlEdgeBottom)

.LineStyle = xlContinuous

.Weight = xlThin

.ColorIndex = xlAutomatic

End With

With Selection.Borders(xlEdgeRight)

.LineStyle = xlContinuous

.Weight = xlThin

.ColorIndex = xlAutomatic

End With

With Selection.Borders(xlInsideVertical)

.LineStyle = xlContinuous

.Weight = xlThin

.ColorIndex = xlAutomatic

End With

With Selection.Borders(xlInsideHorizontal)

.LineStyle = xlContinuous

.Weight = xlThin

.ColorIndex = xlAutomatic

End With

Row = Row + k + 5

End If

End If

j = j + 1

Loop Until Len(Value) = 0

Next i

End With

End Sub

Private Sub Down\_Click()

Dim Index As Integer

Dim Value As String

Index = Queue.ListIndex

If Index <> Queue.ListCount - 1 And Index <> - 1 Then

Value = Queue.Value

Queue.RemoveItem (Index)

Queue.AddItem Value, Index + 1

Queue.ListIndex = Index + 1

End If

End Sub

Private Sub Group\_Change()

Dim i As Integer

Dim Value As String

i = 2

Student.Clear

With ActiveWorkbook.Sheets(1)

Do

Value = .Cells(i, 1).Value

If Value = Group.Value Then

Student.AddItem .Cells(i, 2).Value + " " + .Cells(i, 3).Value + " " + .Cells(i, 4).Value

End If

i = i + 1

Loop Until Len(Value) = 0

End With

Student.ListIndex = 0

End Sub

Private Sub Remove\_Click()

If Not (Queue.ListCount = 0 Or Queue.ListIndex = - 1) Then

Queue.RemoveItem Queue.ListIndex

End If

End Sub

Private Sub Up\_Click()

Dim Index As Integer

Dim Value As String

Index = Queue.ListIndex

If Index > 0 Then

Value = Queue.Value

Queue.RemoveItem Index

Queue.AddItem Value, Index - 1

Queue.ListIndex = Index - 1

End If

End Sub

Private Sub UserForm\_Initialize()

Dim i As Integer

Dim Value As String

i = 2

With ActiveWorkbook.Sheets(1)

Do

Value = .Cells(i, 1).Value

If Len(Value) > 0 Then

For j = 0 To Group.ListCount - 1

If Group.List(j) = Value Then Exit For

Next j

If j = Group.ListCount Then Group.AddItem Value

i = i + 1

End If

Loop Until Len(Value) = 0

End With

Group.ListIndex = 0

End Sub

**Форма «DataForm»:**

Dim Cursor As Integer

Dim Count As Integer

Dim ObjectList() As Integer

Private Sub AutumnCtrl\_Change()

Autumn.Value = AutumnCtrl.Value

ObjectList(Objects.ListIndex, 0) = Autumn.Value

End Sub

Private Sub Backward\_Click()

If Cursor - 1 >= 0 Then

Cursor = Cursor - 1

LoadObject (Cursor)

End If

End Sub

Private Sub Begin\_Click()

Cursor = 0

LoadObject (Cursor)

End Sub

Private Sub Cancel\_Click()

Me.Hide

End Sub

Private Sub Delete\_Click()

If Cursor <> Count Then

ActiveWorkbook.Sheets("Storage").Rows(Cursor + 2).Delete Shift:=xlUp

If Cursor = Count - 1 Then Cursor = Cursor - 1

Count = Count - 1

LoadObject (Cursor)

End If

End Sub

Private Sub Forward\_Click()

If Cursor + 1 <= Count Then

If Cursor + 1 = Count Then

NewItem\_Click

Exit Sub

End If

Cursor = Cursor + 1

LoadObject (Cursor)

End If

End Sub

Private Sub NewItem\_Click()

Cursor = Count

ClearObject

ItemCount.Caption = "Новый элемент"

End Sub

Private Sub Objects\_Change()

Autumn.Value = ObjectList(Objects.ListIndex, 0)

Spring.Value = ObjectList(Objects.ListIndex, 1)

AutumnCtrl.Value = Autumn.Value

SpringCtrl.Value = Spring.Value

End Sub

Private Sub SpringCtrl\_Change()

Spring.Value = SpringCtrl.Value

ObjectList(Objects.ListIndex, 1) = Spring.Value

End Sub

Private Sub Update\_Click()

With ActiveWorkbook.Sheets("Storage")

If (Len(Group.Value) = 0) Then

MsgBox "Необходимо ввести номер группы!"

Exit Sub

End If

If (Len(Last.Value) = 0) Then

MsgBox "Необходимо ввести фамилию студента!"

Exit Sub

End If

If (Len(First.Value) = 0) Then

MsgBox "Необходимо ввести имя студента!"

Exit Sub

End If

If (Len(Middle.Value) = 0) Then

MsgBox "Необходимо ввести отчество студента!"

Exit Sub

End If

.Cells(Cursor + 2, 1) = Group.Value

.Cells(Cursor + 2, 2) = Last.Value

.Cells(Cursor + 2, 3) = First.Value

.Cells(Cursor + 2, 4) = Middle.Value

For i = 0 To Objects.ListCount - 1

.Cells(Cursor + 2, i + 5) = Trim(Str(ObjectList(i, 0))) + ":" + Trim(Str(ObjectList(i, 1)))

Next i

End With

If Cursor = Count Then

Count = Count + 1

ItemCount.Caption = "Элемент " + Str(Cursor + 1) + " из " + Str(Count)

End If

End Sub

Private Sub UserForm\_Initialize()

Dim i As Integer

Dim Value As String

Cursor = 0

i = 5

Count = GetListCount()

With ActiveWorkbook.Sheets("Storage")

Do

Value = .Cells(1, i).Value

If Len(Value) > 0 Then

For j = 0 To Objects.ListCount - 1

If Objects.List(j) = Value Then Exit For

Next j

If j = Objects.ListCount Then Objects.AddItem Value

i = i + 1

End If

Loop Until Len(Value) = 0

End With

ReDim ObjectList(Objects.ListCount - 1, 1)

LoadObject (Cursor)

End Sub

Private Sub LoadObject(Index As Integer)

With ActiveWorkbook.Sheets("Storage")

Group.Value = .Cells(Cursor + 2, 1)

Last.Value = .Cells(Cursor + 2, 2)

First.Value = .Cells(Cursor + 2, 3)

Middle.Value = .Cells(Cursor + 2, 4)

For i = 0 To Objects.ListCount - 1

ObjectList(i, 0) = Split(.Cells(Cursor + 2, i + 5), ":")(0)

ObjectList(i, 1) = Split(.Cells(Cursor + 2, i + 5), ":")(1)

Next i

If Objects.ListIndex = 0 Then Call Objects\_Change

Objects.ListIndex = 0

ItemCount.Caption = "Элемент " + Str(Cursor + 1) + " из " + Str(Count)

End With

End Sub

Private Function GetListCount() As Integer

Dim i As Integer

GetListCount = 0

While Len(ThisWorkbook.Sheets("Storage").Cells(GetListCount + 2, 1).Value) <> 0

GetListCount = GetListCount + 1

Wend

End Function

Private Sub ClearObject()

Group.Value = ""

First.Value = ""

Last.Value = ""

Middle.Value = ""

For i = 0 To Objects.ListCount - 1

ObjectList(i, 0) = 2

ObjectList(i, 1) = 2

Next i

Autumn.Value = ""

Spring.Value = ""

Call Objects\_Change

End Sub

**Форма «DiaGroupForm»:**

Private Sub Autumn\_Click()

Autumn.Value = True

Spring.Value = False

End Sub

Private Sub Cancel\_Click()

Me.Hide

End Sub

Private Sub Create\_Click()

Dim j As Integer

Dim Stat(3) As Integer

Dim Index As Integer

Dim Count As Integer

Dim ChartObj As ChartObject

j = 2

Index = 0

For i = 0 To 3

Stat(i) = 0

Next i

If Autumn.Value = True Then Index = 0 Else Index = 1

ActiveWorkbook.Worksheets.Add After:=Worksheets(Worksheets.Count)

With ActiveSheet

.Cells.Clear

Do

Value = Worksheets("Storage").Cells(j, 1).Value

If Len(Value) > 0 Then

If Worksheets("Storage").Cells(j, 1).Value = Group.Value Then

If Split(Worksheets("Storage").Cells(j, Object.ListIndex + 5).Value, ":")(Index) = "5" Then Stat(0) = Stat(0) + 1

If Split(Worksheets("Storage").Cells(j, Object.ListIndex + 5).Value, ":")(Index) = "4" Then Stat(1) = Stat(1) + 1

If Split(Worksheets("Storage").Cells(j, Object.ListIndex + 5).Value, ":")(Index) = "3" Then Stat(2) = Stat(2) + 1

If Split(Worksheets("Storage").Cells(j, Object.ListIndex + 5).Value, ":")(Index) = "2" Then Stat(3) = Stat(3) + 1

End If

End If

j = j + 1

Loop Until Len(Value) = 0

.Cells(1, 1).Value = "Диаграмма успеваемости группы " + Group.Value

.Cells(2, 1).Value = "по предмету " + Object.Value

.Cells(3, 1).Value = "за " + Str(Index + 1) + "-й семестр"

For i = 0 To 3

.Cells(i + 4, 1).Value = "Оценка " + Str(5 - i)

.Cells(i + 4, 2).Value = Str(Stat(i))

Next i

Set ChartObj = .ChartObjects.Add(0, 0, 400, 400)

ChartObj.Chart.ChartType = xlPie

ChartObj.Chart.SeriesCollection.Add Source:=Range(.Cells(4, 2), .Cells(7, 2))

ChartObj.Chart.SeriesCollection.Item(1).XValues = Range(.Cells(4, 1), .Cells(7, 1))

ChartObj.Chart.HasLegend = True

ChartObj.Chart.Legend.Position = xlBottom

ChartObj.Chart.HasTitle = True

ChartObj.Chart.ChartTitle.Text = "Диаграмма успеваемости группы " + Group.Value + Chr(10) + "по предмету " + Object.Value + Chr(10) + "за " + Str(Index + 1) + "-й семестр"

End With

End Sub

Private Sub Spring\_Click()

Spring.Value = True

Autumn.Value = False

End Sub

Private Sub UserForm\_Initialize()

Dim i As Integer

Dim Value As String

i = 2

With ActiveWorkbook.Sheets(1)

Do

Value = .Cells(i, 1).Value

If Len(Value) > 0 Then

For j = 0 To Group.ListCount - 1

If Group.List(j) = Value Then Exit For

Next j

If j = Group.ListCount Then Group.AddItem Value

i = i + 1

End If

Loop Until Len(Value) = 0

End With

Group.ListIndex = 0

For j = 0 To 8

Object.AddItem ActiveWorkbook.Sheets("Storage").Cells(1, j + 5).Value

Next j

Object.ListIndex = 0

Autumn.Value = True

End Sub

**Форма «DiaStudentForm»:**

Private Sub Autumn\_Click()

Autumn.Value = True

Spring.Value = False

End Sub

Private Sub Cancel\_Click()

Me.Hide

End Sub

Private Sub Create\_Click()

Dim j As Integer

Dim Stat(3) As Integer

Dim Index As Integer

Dim Count As Integer

Dim ChartObj As ChartObject

j = 2

Index = 0

For i = 0 To 3

Stat(i) = 0

Next i

If Autumn.Value = True Then Index = 0 Else Index = 1

ActiveWorkbook.Worksheets.Add After:=Worksheets(Worksheets.Count)

With ActiveSheet

.Cells.Clear

Do

Value = Worksheets("Storage").Cells(j, 1).Value

If Len(Value) > 0 Then

If Worksheets("Storage").Cells(j, 2).Value + " " + Worksheets("Storage").Cells(j, 3).Value + " " + Worksheets("Storage").Cells(j, 4).Value = Student.Value Then Exit Do

End If

j = j + 1

Loop Until Len(Value) = 0

For i = 0 To 8

If Split(Worksheets("Storage").Cells(j, i + 5).Value, ":")(Index) = "5" Then Stat(0) = Stat(0) + 1

If Split(Worksheets("Storage").Cells(j, i + 5).Value, ":")(Index) = "4" Then Stat(1) = Stat(1) + 1

If Split(Worksheets("Storage").Cells(j, i + 5).Value, ":")(Index) = "3" Then Stat(2) = Stat(2) + 1

If Split(Worksheets("Storage").Cells(j, i + 5).Value, ":")(Index) = "2" Then Stat(3) = Stat(3) + 1

Next i

.Cells(1, 1).Value = "Диаграмма успеваемости студента: "

.Cells(2, 1).Value = Student.Value

.Cells(3, 1).Value = Str(Index + 1) + "-й семестр"

For i = 0 To 3

.Cells(i + 4, 1).Value = "Оценка " + Str(5 - i)

.Cells(i + 4, 2).Value = Str(Stat(i))

Next i

Set ChartObj = .ChartObjects.Add(0, 0, 400, 400)

ChartObj.Chart.ChartType = xlPie

ChartObj.Chart.SeriesCollection.Add Source:=Range(.Cells(4, 2), .Cells(7, 2))

ChartObj.Chart.SeriesCollection.Item(1).XValues = Range(.Cells(4, 1), .Cells(7, 1))

ChartObj.Chart.HasLegend = True

ChartObj.Chart.Legend.Position = xlBottom

ChartObj.Chart.HasTitle = True

ChartObj.Chart.ChartTitle.Text = "Диаграмма успеваемости студента" + Chr(10) + Student.Value + Chr(10) + Str(Index + 1) + "-й семестр"

End With

End Sub

Private Sub Group\_Change()

Dim i As Integer

Dim Value As String

i = 2

Student.Clear

With ActiveWorkbook.Sheets("Storage")

Do

Value = .Cells(i, 1).Value

If Value = Group.Value Then

Student.AddItem .Cells(i, 2).Value + " " + .Cells(i, 3).Value + " " + .Cells(i, 4).Value

End If

i = i + 1

Loop Until Len(Value) = 0

End With

Student.ListIndex = 0

End Sub

Private Sub Spring\_Click()

Spring.Value = True

Autumn.Value = False

End Sub

Private Sub UserForm\_Initialize()

Dim i As Integer

Dim Value As String

i = 2

With ActiveWorkbook.Sheets("Storage")

Do

Value = .Cells(i, 1).Value

If Len(Value) > 0 Then

For j = 0 To Group.ListCount - 1

If Group.List(j) = Value Then Exit For

Next j

If j = Group.ListCount Then Group.AddItem Value

i = i + 1

End If

Loop Until Len(Value) = 0

End With

Group.ListIndex = 0

Autumn.Value = True

End Sub

**Модули:**

**Модуль Main:**

Public Sub QDialog1()

MainForm.Show

End Sub

Модуль Module 1:

Sub Macro1()

'' Macro1 Macro

' Macro recorded 25.12.2006 by Zeon

'Columns("A:A").ColumnWidth = 33.29

Columns("B:B").ColumnWidth = 17.14

Columns("C:C").ColumnWidth = 16.57

End Sub

**Модуль Module 2:**

Sub Macro2()

'' Macro2 Macro

' Macro recorded 25.12.2006 by Zeon

'Range("C17:D17").Select

With Selection

.HorizontalAlignment = xlCenter

.VerticalAlignment = xlBottom

.WrapText = False

.Orientation = 0

.AddIndent = False

.IndentLevel = 0

.ShrinkToFit = False

.ReadingOrder = xlContext

.MergeCells = False

End With

Selection.Merge

Range("C16:D16").Select

With Selection

.HorizontalAlignment = xlCenter

.VerticalAlignment = xlBottom

.WrapText = False

.Orientation = 0

.AddIndent = False

.IndentLevel = 0

.ShrinkToFit = False

.ReadingOrder = xlContext

.MergeCells = False

End With

Selection.Merge

Range("C19:D27").Select

With Selection

.HorizontalAlignment = xlCenter

.VerticalAlignment = xlBottom

.WrapText = False

.Orientation = 0

.AddIndent = False

.IndentLevel = 0

.ShrinkToFit = False

.ReadingOrder = xlContext

.MergeCells = False

End With

Range("B16:D27").Select

Selection.Borders(xlDiagonalDown).LineStyle = xlNone

Selection.Borders(xlDiagonalUp).LineStyle = xlNone

With Selection.Borders(xlEdgeLeft)

.LineStyle = xlContinuous

.Weight = xlThin

.ColorIndex = xlAutomatic

End With

With Selection.Borders(xlEdgeTop)

.LineStyle = xlContinuous

.Weight = xlThin

.ColorIndex = xlAutomatic

End With

With Selection.Borders(xlEdgeBottom)

.LineStyle = xlContinuous

.Weight = xlThin

.ColorIndex = xlAutomatic

End With

With Selection.Borders(xlEdgeRight)

.LineStyle = xlContinuous

.Weight = xlThin

.ColorIndex = xlAutomatic

End With

With Selection.Borders(xlInsideVertical)

.LineStyle = xlContinuous

.Weight = xlThin

.ColorIndex = xlAutomatic

End With

With Selection.Borders(xlInsideHorizontal)

.LineStyle = xlContinuous

.Weight = xlThin

.ColorIndex = xlAutomatic

End With

Range("C16:D17").Select

With Selection

.HorizontalAlignment = xlLeft

.VerticalAlignment = xlBottom

.WrapText = False

.Orientation = 0

.AddIndent = False

.IndentLevel = 0

.ShrinkToFit = False

.ReadingOrder = xlContext

End With

Range("B16:B27").Select

With Selection.Interior

.ColorIndex = 36

.Pattern = xlSolid

End With

Range("C16:D17").Select

With Selection.Interior

.ColorIndex = 35

.Pattern = xlSolid

End With

Range("C18:D18").Select

With Selection.Interior

.ColorIndex = 37

.Pattern = xlSolid

End With

Range("C19:D27").Select

With Selection.Interior

.ColorIndex = 40

.Pattern = xlSolid

End With

End Sub