## Процедура поиска и удаления пустых строк

Sub DeleteEmptyRows()

    LastRow = ActiveSheet.UsedRange.Row - 1 + ActiveSheet.UsedRange.Rows.Count    'определяем размеры таблицы

    Application.ScreenUpdating = False

    For r = LastRow To 1 Step -1           'проходим от последней строки до первой

        If Application.CountA(Rows(r)) = 0 Then Rows(r).Delete   'если в строке пусто - удаляем ее

    Next r

End Sub

## Находит определенное значение и закрашивает всю строку

Sub FindAndSelect()

Dim Rng As Range

Dim n

Dim rData

'Dim RndMY

With Worksheets("Лист1")

Set Rng = .Range("A2:G100")

For Each n In Rng

Select Case n

Case "уфа"

Range(Cells(n.Row, 1), Cells(n.Row, 7)).Interior.Color = RGB(146, 208, 80)

Case "бирск"

Range(Cells(n.Row, 1), Cells(n.Row, 7)).Interior.Color = RGB(255, 255, 0)

End Select

Next n

End With

End Sub

## Выбрать отдельные части таблицы

Sub SelectingPartOfTable()

Dim oSh As Worksheet

Set oSh = ActiveSheet

'1: работаем с таблицей целиком

With oSh.ListObjects("Таблица1")

MsgBox .Name

'выбрать всю таблицу

.Range.Select

'выбарть только данные

.DataBodyRange.Select

'выбрать третью колонку

.ListColumns(3).Range.Select

'выбрать только первую колонку

.ListColumns(1).DataBodyRange.Select

'выбрать только четвертый столбец (заголовки таблицы не считаются)

.ListRows(4).Range.Select

End With

'2: работаем с выделением

'выбрать данные в столбце без заголовков

oSh.Range("Таблица1[Столбец2]").Select

'выбрать данные и заголовок

oSh.Range("Таблица1[[#All],[Столбец1]]").Select

'выбрать все данные в таблице

oSh.Range("Таблица1").Select

'выбрать всю таблицу

oSh.Range("Таблица1[#All]").Select

'выбрать одну строку

oSh.Range("A5:F5").Select

End Sub

## Процедура выбирает и подкрашивает 10 наибольших значений

Sub TopTen()

Selection.FormatConditions.AddTop10

Selection.FormatConditions(Selection.FormatConditions.Count).SetFirstPriority

With Selection.FormatConditions(1)

.TopBottom = xlTop10Top

.Rank = 10

.Percent = False

End With

With Selection.FormatConditions(1).Font

.Color = -16752384

.TintAndShade = 0

End With

With Selection.FormatConditions(1).Interior

.PatternColorIndex = xlAutomatic

.Color = 13561798

.TintAndShade = 0

End With

Selection.FormatConditions(1).StopIfTrue = False

End Sub

## Выделяет занчения больше заданного

Sub HighlightGreaterThanValues()

Dim i As Integer

i = InputBox("Enter Greater Than Value", "Enter Value")

Selection.FormatConditions.Delete

Selection.FormatConditions.Add Type:=xlCellValue, Operator:=xlGreater, Formula1:=i

'Selection.FormatConditions.Add Type:=xlCellValue, Operator:=xlLower, Formula1:=i

Selection.FormatConditions(Selection.FormatConditions.Count).SetFirstPriority

With Selection.FormatConditions(1)

.Font.Color = RGB(0, 0, 0)

.Interior.Color = RGB(31, 218, 154)

End With

End Sub

## Выбирает отрицательные значения

Sub highlightNegativeNumbers()  
Dim Rng As Range  
For Each Rng In Selection  
If WorksheetFunction.IsNumber(Rng) Then  
If Rng.Value < 0 Then  
Rng.Font.Color= -16776961  
End If  
End If  
Next  
End Sub

## Выделяет уникальные значения

Sub highlightUniqueValues()

Dim rng As Range

Set rng = Selection

rng.FormatConditions.Delete

Dim uv As UniqueValues

Set uv = rng.FormatConditions.AddUniqueValues

uv.DupeUnique = xlUnique

uv.Interior.Color = vbGreen

End Sub

## Перенос данных вместе с заголовками

Sub CopyFilteredRowsAndHeadingsList1()

Dim ws As Worksheet

Dim rng As Range

Dim rng2 As Range

Dim Lst As ListObject

Application.ScreenUpdating = False

Set Lst = ActiveWorkbook.Worksheets("Лист4").ListObjects("Таблица4")

With Lst.AutoFilter.Range

On Error Resume Next

Set rng2 = .Offset(1, 0).Resize(.Rows.Count - 1, 1) \_

.SpecialCells(xlCellTypeVisible)

On Error GoTo 0

End With

If rng2 Is Nothing Then

MsgBox "No data to copy"

Else

Set ws = Sheets.Add

Set rng = Lst.AutoFilter.Range

'copy rows with headings

rng.SpecialCells(xlCellTypeVisible).Copy \_

Destination:=ws.Range("A1")

End If

Application.ScreenUpdating = True

End Sub

## Комбинируем данные с разных листов

Процедура, которая комбинирует данные с разных листов. Количество столбцов одинаковое, названия тоже. Данные начинаются с ячейки А1, иначе не сработает. Работает только если все рабочие листы в одной рабочей книге

Sub Combine()

Dim i As Integer

Dim xTCount As Variant

Dim xWs As Worksheet

On Error Resume Next

LInput:

xTCount = Application.InputBox("The number of title rows", "", "1")

If TypeName(xTCount) = "Boolean" Then Exit Sub

If Not IsNumeric(xTCount) Then

MsgBox "Only can enter number", , "Kutools for Excel"

GoTo LInput

End If

Set xWs = ActiveWorkbook.Worksheets.Add(Sheets(1))

xWs.Name = "Combined"

Worksheets(2).Range("A1").EntireRow.Copy Destination:=xWs.Range("A1")

For i = 2 To Worksheets.Count

Worksheets(i).Range("A1").CurrentRegion.Offset(CInt(xTCount), 0).Copy \_

Destination:=xWs.Cells(xWs.UsedRange.Cells(xWs.UsedRange.Count).Row + 1, 1)

Next

End Sub

## Переносим в активную книгу листы из другой книги

Процедура, которая переносит все листы из другой книги. Она запрашивает имя файл и вставляет все листы из другой книги в активную.

Sub MergeExcelFiles()

Dim fnameList, fnameCurFile As Variant

Dim countFiles, countSheets As Integer

Dim wksCurSheet As Worksheet

Dim wbkCurBook, wbkSrcBook As Workbook

fnameList = Application.GetOpenFilename(FileFilter:="Microsoft Excel Workbooks (\*.xls;\*.xlsx;\*.xlsm),\*.xls;\*.xlsx;\*.xlsm", Title:="Choose Excel files to merge", MultiSelect:=True)

If (vbBoolean <> VarType(fnameList)) Then

If (UBound(fnameList) > 0) Then

countFiles = 0

countSheets = 0

Application.ScreenUpdating = False

Application.Calculation = xlCalculationManual

Set wbkCurBook = ActiveWorkbook

For Each fnameCurFile In fnameList

countFiles = countFiles + 1

Set wbkSrcBook = Workbooks.Open(Filename:=fnameCurFile)

For Each wksCurSheet In wbkSrcBook.Sheets

countSheets = countSheets + 1

wksCurSheet.Copy after:=wbkCurBook.Sheets(wbkCurBook.Sheets.Count)

Next

wbkSrcBook.Close SaveChanges:=False

Next

Application.ScreenUpdating = True

Application.Calculation = xlCalculationAutomatic

MsgBox "Processed " & countFiles & " files" & vbCrLf & "Merged " & countSheets & " worksheets", Title:="Merge Excel files"

End If

Else

MsgBox "No files selected", Title:="Merge Excel files"

End If

End Sub

## Разделяем книгу на отдельные листы

Sub SplitSheets()

Dim wbSource As Workbook, wbDest As Workbook

Dim ws As Worksheet

Dim sPath1 As String, sPath2 As String

If ActiveWorkbook Is Nothing Then

Call MsgBox("You need an Active Workbook", vbCritical + vbOKOnly, "Split Workbook")

Exit Sub

End If

Set wbSource = ActiveWorkbook '<<<<<<<<<<<<<<<<<<

sPath1 = wbSource.Path

Application.ScreenUpdating = False

For Each ws In wbSource.Worksheets

If ws.Visible Then

ws.Copy

Set wbDest = ActiveWorkbook

sPath2 = sPath1 & Application.PathSeparator & ws.Range("A1").Text

' sPath2 = sPath1 & Application.PathSeparator & ws.Name

On Error Resume Next

Kill sPath2 & ".xlsx"

On Error GoTo 0

On Error GoTo CanNotSaveIt

Call wbDest.SaveAs(sPath2, xlOpenXMLWorkbook)

Call wbDest.Close(False)

End If

Next

wbSource.Activate

Application.ScreenUpdating = False

Exit Sub

CanNotSaveIt:

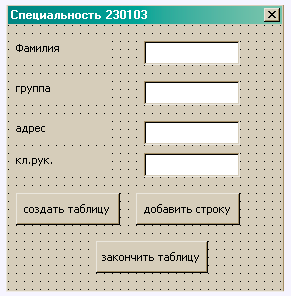
Call MsgBox("Can not save" & vbCrLf & vbCrLf & sPath2, vbCritical + vbOKOnly, "Split Workbook")

Resume Next

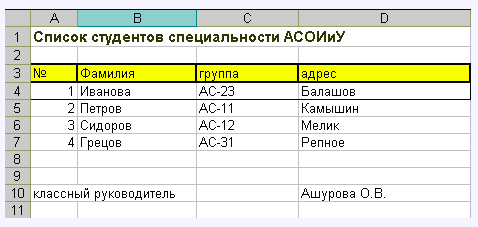
End Sub

## Создать программу, которая по введенным текстовым данным в соотвествующие текстовые поля формы автоматизирует ввод данных на студентов некоторой специальности учебного заведения. Результаты заполнения текстовых полей выводятся на лист excel, что позволяет при необходимости распечатать данные.

1. Создать форму



Подготовить лист эксель



4. Обработайте кнопки.

Кнопка Создать таблицу

Const strNomer = 3 количество строк для заголовка

Dim strName1 As String 'строка для адресации ячеек

Dim strName2 As String

Dim nomer As Long номер очередной строки таблицы

Private Sub CommandButton1\_Click()

ActiveWorkbook.SaveAs («работа с базой данных. xls»)

nomer = 1 End Sub

Кнопка Добавить строку

Private Sub CommandButton2\_Click()

strName1 = Trim(Str(strNomer + nomer))

With ActiveSheet 'ввод данных для новой отчетной таблицы

Range("A" + strName1).Value = nomer

Range("B" + strName1).Value = TextBox1.Text

Range("C" + strName1).Value = TextBox2.Text

Range("D" + strName1).Value = TextBox3.Text

'автозаполнение с текущей строки таблицы

strName2 = Trim(Str(strNomer + nomer + 1))

Set range1 =.Range("A" + strName1 +":D" + strName1)

Set range2 =.Range("A" + strName1 +":D" + strName2)

range1.AutoFill Destination:=range2

Range("A" + strName2 +":D" + strName2).Clear

End With

' очистка полей формы для ввода очередной записи

TextBox1.Text = ""

TextBox2.Text = ""

TextBox3.Text = ""

TextBox1.SetFocus

nomer = nomer + 1

End Sub

Кнопка Закончить таблицу

Private Sub CommandButton3\_Click()

'закрытие формы подведение итогов и вывод фамилии преподавателя

UserForm1.Hide

With ActiveSheet

strName2 = Trim(Str(strNomer + nomer + 2))

Range("A" + strName2).Value = «классный руководитель»

Range("D" + strName2).Value = TextBox4.Text

End With

End Sub