

Odstranění řádků (na základě hodnoty) patří k častým dotazům v diskusích. Úlohu je třeba rozdělit na dvě části - nalezení požadovaných buněk a následné odmazání řádků, na kterých leží.

Výběr buněk lze provádět několika způsoby - filtrem, s pomocí dialogů Najít a nahradit (CTRL + F) či Přejít na/Jinak (F5, CTRL + G, tlačítko Jinak), resp. karta Domů / skupina Úpravy, Najít a vybrat.



Příklad 1 - prázdné buňky jako kritérium

Filtrování



Automatický filtr

Tip: Ve vlastním filtru je možné používat zástupné symboly * a ?.

* ... žádný, jeden nebo více znaků

? ... jeden libovolný znak

P* ... textový řetězec začínající písmenem P

olo ... textový řetězec obsahující „olo“ (na začátku, uprostřed, na konci)

????????? ... devítiznakový řetězec

Dialog Najít a nahradit



Dialog Najít a nahradit / Najít vše / CTRL + A

Podstatné je ve výsledcích vyhledávání (tlačítko Najít vše) vybrat všechny položky ve výpisu (CTRL + A). Tím dojde k vyznačení buněk i na listu.

Tip: I v dialogu Najít a nahradit je možné používat zástupné symboly * a ?. Kromě toho zvládá vyhledávat i podle formátu (rozklepněte tlačítko Možnosti). Tímto způsobem mimochodem dokážete

vybrat i odemčené buňky.

Dialog Přejít na – jinak



Dialog Přejít na – jinak

Dialog Přejít na – jinak vhodně doplňuje vyhledávání. Jeho schopnosti ve VBA představuje metoda SpecialCells, jak si ukážeme později.

Jsou-li vybrány buňky a je potřeba odstranit celé řádky, stačí v kontextovém menu vybrat Odstranit / Celý řádek.



Volba Odstranit / Celý řádek

Pozn. Je škoda, že nefunguje klávesová zkratka SHIFT + mezerník. Ta vybírá celý řádek jen pro aktivní buňku.

Odstranění řádků ve VBA



Příklad 1 – prázdné buňky jako kritérium

První ukázky kódu prochází oblast položku po položce shora dolů. Testují, zda-li je buňka prázdná (IsEmpty) a pokud ano, odmazávají celý řádek (EntireRow.Delete). A obě jsou špatně...

```
1 Sub OdstraneniRadkuA()  
2  
3 Dim rngOblast As Range  
4 Dim rngBunka As Range  
5  
6 For Each rngBunka In Range("B2:B7")  
7  
8 If IsEmpty(rngBunka) Then  
9  
10 rngBunka.EntireRow.Delete  
11  
12 End If  
13  
14 Next rngBunka  
15  
16 End Sub
```

```
1 Sub OdstraneniRadkuB()  
2  
3 Dim rngOblast As Range  
4 Dim rngBunka As Range  
5  
6 Set rngOblast = Range("B2:B7")  
7  
8 For Each rngBunka In rngOblast  
9  
10 If IsEmpty(rngBunka) Then  
11  
12 rngBunka.EntireRow.Delete  
13  
14 End If  
15  
16 Next rngBunka  
17  
18 End Sub
```

Procedury jsou si velmi podobné a obsahují programátorskou tfuajtajblovnost. Neplatí zde pravidlo, že při průchodu cyklem nesmí programátor zasahovat do čítače, resp. jeho mezí (odstranění řádku mění prohledávanou oblast). Pokud tak činíte, pak jen ve zvláštních případech a při velmi dobře promyšleném algoritmu.

Předkládám ještě jednu proceduru prohledávání shora dolů. Tentokrát si ji odkrokujte (okna vedle sebe, F8). I z tohoto důvodu je zde řádek rngBunka.Select.

1	Sub OdstraneniRadkuC()
2	
3	Dim rngOblast As Range
4	Dim rngBunka As Range
5	
6	Dim i As Integer
7	Dim intPocet As Integer
8	
9	Set rngOblast = Range("B2:B7")
10	
11	intPocet = rngOblast.Cells.Count
12	
13	For i = 1 To intPocet
14	
15	Set rngBunka = rngOblast.Cells(i)
16	
17	rngBunka. Select
18	
19	If IsEmpty(rngBunka) Then
20	
21	rngBunka.EntireRow.Delete
22	
23	End If
24	
25	Next i
26	
27	End Sub

Kód zde sice neobsahuje chybu uvedenou výše, ale přeci jen je postaven na prapodivném základu. Zatímco čítač hodnoty navyšuje, oblast se zmenšuje... Je to jako řídit auto v protisměru. Zapomeňte i na ni.

Odstraňování řádků je dost možná jediný případ, kdy je vhodné cyklus obrátit a procházet položky od konce. Doporučuji kód odkrokovat.

1	Sub OdstraneniRadkuD()
2	
3	Dim rngOblast As Range
4	Dim rngBunka As Range
5	
6	Dim i As Integer
7	Dim intPocet As Integer
8	
9	Set rngOblast = Range("B2:B7")
10	
11	intPocet = rngOblast.Cells.Count
12	
13	For i = intPocet To 1 Step -1
14	
15	Set rngBunka = rngOblast.Cells(i)
16	
17	rngBunka. Select
18	
19	If IsEmpty(rngBunka) Then
20	
21	rngBunka.EntireRow.Delete
22	
23	End If
24	
25	Next i
26	
27	End Sub

Jak bylo napsáno, dialog Přejít na – jinak ve pod VBA zastoupen metódou SpecialCells. V případě prázdných buněk tak můžeme vypustit cyklus a kód se smrskne do jednoho řádku.

1	Sub OdstraneniRadkuE()
2	
3	Dim rngOblast As Range
4	
5	Set rngOblast = Range("B2:B7")
6	
7	rngOblast.SpecialCells(xlCellTypeBlanks).EntireRow.Delete
8	
9	End Sub

A jak to udělat v případě, že kritériem pro odstranění řádků nebudou prázdné buňky, ale konkrétní hodnota (hodnoty)?



Příklad 2

Podle obrázku je naším cílem vymazat řádky s hodnotami začínajícími písmenem X. Vrátime se k dialogu Najít a nahradit. Pod VBA není bohužel možné nasimulovat chování Najít vše / CTRL + A. Nicméně jeho prostřednictvím si připravíme půdu pro SpecialCells.

```
1  Sub OdstraneniRadkuF()  
2  
3  Dim rngOblast As Range  
4  
5  Set rngOblast = Range("B2:B7")  
6  
7  With rngOblast  
8  
9      .Replace What:="X*", Replacement:="=NA()", LookAt:=xlPart, _  
10     SearchOrder:=xlByRows, MatchCase:=False, SearchFormat:=False, _  
11     ReplaceFormat:=False  
12  
13     .SpecialCells(xlCellTypeFormulas, xlErrors).EntireRow.Delete  
14  
15 End With  
16  
17 End Sub
```

Uvedené makro nahrazuje položky začínající písmenem X za vzorec =NA(), což se v českém prostředí na listu objeví jako =NEDEF(). Výsledek této funkce (chybová hodnota #NENÍ K DISPOZICI často využívaná v grafech) následně zpracuje metoda SpecialCells. Snímek níže ukazuje, jak by vypadal výběr chybových hodnot ručně.



Dialog Přejít na - jinak (chybové hodnoty)

Příloha

[odstraneni_radku.zip](#)