

ArrayList představuje třídu .NET Frameworku ([System.Collections.Arraylist](#), knihovna mscorlib). Dokáže to, co musíme jinak horko těžko při práci s prostým polem ve VBA programovat - třídění, přidávání a odebírání položek, test existence položky, spojování polí aj. Vtip je v tom, že s jistými omezeními je třída ArrayList dostupná i pod VBA. Musím přiznat, že na tuto skutečnost jsem narazil poprvé cca před rokem a půl. Žádná další teorie nás nečeká, pojďme si zkusit pár příkladů. Pozn. Doporučuji kód nekrokovat, ale spouštět až do místa vložené záložky. I tak budete muset leckdy pro zobrazení skutečného obsahu proměnné v okně Locals ji sbalit a znovu rozbalit.

## ArrayList v příkladech

```
Sub ProceduraArrayList()
```

```
Dim objArray
```

```
Dim objArrayList1
```

```
Dim objArrayList2
```

```
Dim objArrayList3
```

```
Dim objArrayList4
```

```
Dim objArrayList5
```

```
Set objArrayList1 = CreateObject("System.Collections.ArrayList")
```

```
Set objArrayList2 = CreateObject("System.Collections.ArrayList")
```

```
Set objArrayList3 = CreateObject("System.Collections.ArrayList")
```

```
Set objArrayList4 = CreateObject("System.Collections.ArrayList")
```

```
Set objArrayList5 = CreateObject("System.Collections.ArrayList")
```

```
Dim intIndex As Integer
```

```
Dim intPocetPolozek As Integer
```

```
Dim blnPolozkaExistuje As Boolean
```

```
'plnění pole 1
```

```
With objArrayList1
```

```
.Add ("žirafa")
```

```
.Add ("čížek")
```

```
.Add ("motýl")
```

```
End With
```

```
'plnění pole 2  
With objArrayList2  
.Add ("pěnkava")  
.Add ("chobotnice")  
End With
```

```
'plnění pole 3  
With objArrayList3  
.Add ("výr")  
.Add ("mravenec")  
End With
```

```
'přidání jednoho pole na konec druhého  
'závorky nutné  
objArrayList1.AddRange (objArrayList2)
```

```
'vložení jednoho pole do druhého na pozici  
'indexy 0, 1, ...  
'závorky nutné  
objArrayList1.InsertRange 1, (objArrayList3)
```

```
'přepsání položek jednoho pole druhým  
'indexy 0, 1, ...  
'závorky nutné  
objArrayList1.SetRange 1, (objArrayList2)
```

```
'odstranění části pole (2 položky od indexu 1)  
objArrayList1.RemoveRange 1, 2
```

```
'přidání položky za n-tou položku  
'0 ... počátek  
objArrayList1.Insert 1, "tučňák"
```

```
'test existence položky  
blnPolozkaExistuje = objArrayList1.Contains("tučňák")
```

```
'index hledané položky s definováním  
'počátku vyhledávání  
'indexy 0, 1, ...  
intIndex = objArrayList1.IndexOf("tučňák", )  
  
'odstranění položky dle jména  
objArrayList1.Remove ("tučňák")  
  
'odstranění položky dle indexu  
'indexy položek 0, 1, ...  
objArrayList1.RemoveAt 1  
  
'vzestupné setřídění položek  
objArrayList1.Sort  
  
'výpis položek do okna Immediate  
Debug.Print Join(objArrayList1.ToArray(), vbCrLf)  
  
'počet položek  
intPocetPolozek = objArrayList1.Count  
  
'přizpůsobení velikosti naplněným položkám  
objArrayList1.TrimToSize  
  
'revers pole  
objArrayList3.Reverse  
  
'kopie ArrayList do obyčejného pole  
objArray = objArrayList3.ToArray()  
  
'převzetí části pole  
'indexy 0, 1, ...  
Set objArrayList4 = objArrayList1.GetRange(1, 2)  
  
'klonování pole
```

```
Set objArrayList5 = objArrayList3.Clone
```

```
'vyčištění pole
```

```
objArrayList1.Clear
```

```
'odstranění proměnných z paměti
```

```
Set objArrayList1 = Nothing
```

```
Set objArrayList2 = Nothing
```

```
Set objArrayList3 = Nothing
```

```
Set objArrayList4 = Nothing
```

```
Set objArrayList5 = Nothing
```

```
End Sub
```

Největší problémy mi dělaly metody SetRange a InsertRange. Podotýkám, že práce s objektovými proměnnými pod VBA vyžaduje příkaz Set, navíc VBA prapodivně pracuje se závorkami u volání. Pokud se pamatuji, prvně jsem ArrayList využil při třídění položek. Metoda Sort vrací korektnější výsledky, než třeba vlastní procedura (technika řazení) QuickSort. Na ukázkou:

čížek

chobotnice

motýl

pěnkava

žirafa

## Reference

Na knihovnu mscorlib.dll je možné se napojit i přes reference a pak není potřeba metody CreateObject. Doporučuji do knihovny nahlédnout přes Object Browser (F2).

```
Sub KnihovnaMSCORLIB()
```

```
'Tools / References / mscorlib.dll
```

```
Dim objArrayList As New ArrayList
```

```
'plnění pole 1
```

```
With objArrayList
```

```
.Add ("žirafa")
```

```
.Add ("čížek")
```

```
.Add ("motýl")
End With
```

```
End Sub
```

Co se mi bohužel nepodařilo zrealizovat, je vytvoření pole s opakováním položek, např.

```
Set objArrayList = ArrayList.Repeat(„abc“, 7)
```

Je velká škoda, že ArrayList se neumí přímo zbavit duplicit. Museli bychom si zkombinovat metodu Contains třeba s GetRange nebo použít jiné objekty (HashSet). Ve VBA už kdysi pro tyto účely navrhl John Walkenbach datový typ vypůjčený z VBScriptu, a to Dictionary (neumožňuje přidání již existující položky v metodě Add). Je-li potřeba v polích pracovat s klíčem a hodnotou, nezapomínejte ani na přímo ve VBA dostupný objekt Collection.

## Zásobník a fronta

V knihovně se lze odvolávat na některé další třídy (System.Collections.Hashtable, System.Collections.SortedList). My se podíváme na zásobník (zbraně), neboli **stack** a frontu (na poště), čili **queue**, známé to typy z algoritmů.

```
Sub ProceduraStack()
```

```
Dim objStack
```

```
'zásobník, také jinak LIFO
```

```
'tj. Last In First Out (poslední dovnitř, první ven)
```

```
Set objStack = CreateObject("System.Collections.Stack")
```

```
'plnění zásobníku
```

```
With objStack
```

```
.Push ("1. náboj")
```

```
.Push ("2. náboj")
```

```
.Push ("3. náboj")
```

```
End With
```

```
'vystřelení 3. náboje
```

```
'tj. naposled přidané položky
```

```
MsgBox objStack.Pop
```

```
'vystřelení 2. náboje  
MsgBox objStack.Pop
```

```
'je v zásobníku 1. náboj?  
MsgBox objStack.Contains("1. náboj")
```

```
'odstranění proměnné z paměti  
Set objStack = Nothing
```

```
End Sub  
Sub ProceduraQueue()
```

```
Dim objQueue
```

```
'fronta, také jinak FIFO  
'tj. First In First Out (první dovnitř, první ven)
```

```
Set objQueue = CreateObject("System.Collections.Queue")
```

```
'plnění fronty  
With objQueue  
.enqueue ("1. člověk")  
.enqueue ("2. člověk")  
.enqueue ("3. člověk")  
End With
```

```
'odchod 1. člověka  
'tj. odebrání první (nejstarší) přidané položky  
MsgBox objQueue.Dequeue
```

```
'odchod 2. člověka  
MsgBox objQueue.Dequeue
```

```
'kdo je na řadě?  
MsgBox objQueue.Peek
```

```
'odstranění proměnné z paměti  
Set objQueue = Nothing
```

End Sub

Příloha

[pole\\_arraylist.zip](#)