

BALTIE 4 – pole, kolekce, zásobníky,..

1. **pole_pozpatku** – v režimu konzole uživatel zadá počet prvků pole, pak zadá číselné prvky pole. Tyto prvky se vypíší, pak se vypíší pozpátku

deklarace pole



zadání hodnot do pole



výpis hodnot pole



výpis hodnot pozpátku



2. **pole_foreach** - tentýž příklad s využitím cyklu foreach v úseku pro výpis hodnot pole

příkaz slouží k postupnému zpracování všech prvků pole

`foreach (lokální_proměnná in název_pole) { příkazy }`



jiný příklad

Příklad (Pro režim 3D)

Do nového pole vloží fialového, červeného a žlutého Baltíka. Poté provede Cyklus se provede tolikrát, kolik je v poli Baltíků.



3. *pole* - nadefinuj pole o 3 prvcích, jejichž hodnoty jsou 10, 20, 30
pak vypiš prvky pole pomocí foreach a dále všechny prvky pole pozpátku

Deklarace pole výčtem prvků

```
// pole [ 3 ] = { 10 , 20 , 30 }
pole [ ] = new int [ 3 ] { 10 , 20 , 30 }
```

jiný způsob deklarace

```
cislo [ ] = { 10 , 20 , 30 }
```

vypis hodnot

```
foreach ( prvek pole ) {
    prvek +
}
```

vypis hodnot pozpátku

```
for ( i = pole.Length - 1 ; i >= 0 ; i -- ) {
    pole [ i ] +
}
```

4. *dvourozmerne_pole*

```
Dvourozměrné pole 3 x 2
matice [ 3 , 2 ]
matice [ 0 , 0 ] = 0
matice [ 1 , 0 ] = 1
matice [ 2 , 0 ] = 2
matice [ 0 , 1 ] = 4
matice [ 1 , 1 ] = 5
matice [ 2 , 1 ] = 6
```

vypis

```
for ( i = 0 ; i < matice.GetLength(1) ; i -- ) {
    for ( i2 = 0 ; i2 < matice.GetLength(0) ; i2 -- ) {
        matice [ i2 , i ] +
    }
}
```

5. *dynamické_pole* - zadej dynamické pole se 4 prvky (10, 20, 30, 40),

Deklarace a vytvoření dynamického pole

`cisla = new ArrayList`

zadání hodnot

`cisla.add(10)`
`cisla.add(20)`
`cisla.add(30)`
`cisla.add(40)`

Deklarace dynamického pole – pro proměnnou vybereme uživatelský typ – práce s kolekcemi – dynamické pole

vypiš prvky pole

`for (prvek : cisla) {`
 `System.out.println(prvek);`
`}`

přidej další číslo - s indexem 2 a hodnotou 100, vypiš prvky pole

`cisla.add(2, 100)`
`for (prvek : cisla) {`
 `System.out.println(prvek);`
`}`

jiný způsob přidání prvku do pole

`cisla.set(2, 100)`

odstraň 1. prvek, vypiš prvky pole

`cisla.remove(0)`
`for (prvek : cisla) {`
 `System.out.println(prvek);`
`}`

zjisti a vypiš počet prvků pole

`System.out.println(cisla.size());`

6. *fronta* – QUEUE skupina prvků, které odebíráme v pořadí, ve kterém je vkládáme (např. vkládáme seshora, odebíráme zespoda, např. tisková fronta,..)
 deklarace fronty, vložení hodnot do fronty, výpis, odstranění prvků

```

deklarace
    deklarace fronta = new Queue
    
```

vložení hodnot do fronty (Enqueue)

```

fronta.Enqueue(15)
fronta.Enqueue(7)
fronta.Enqueue(25)
    
```

výpis prvků fronty

```

foreach (prvek fronta)
    Console.WriteLine(prvek)
    
```

odstranění 1. prvku z fronty (metoda Dequeue)

```

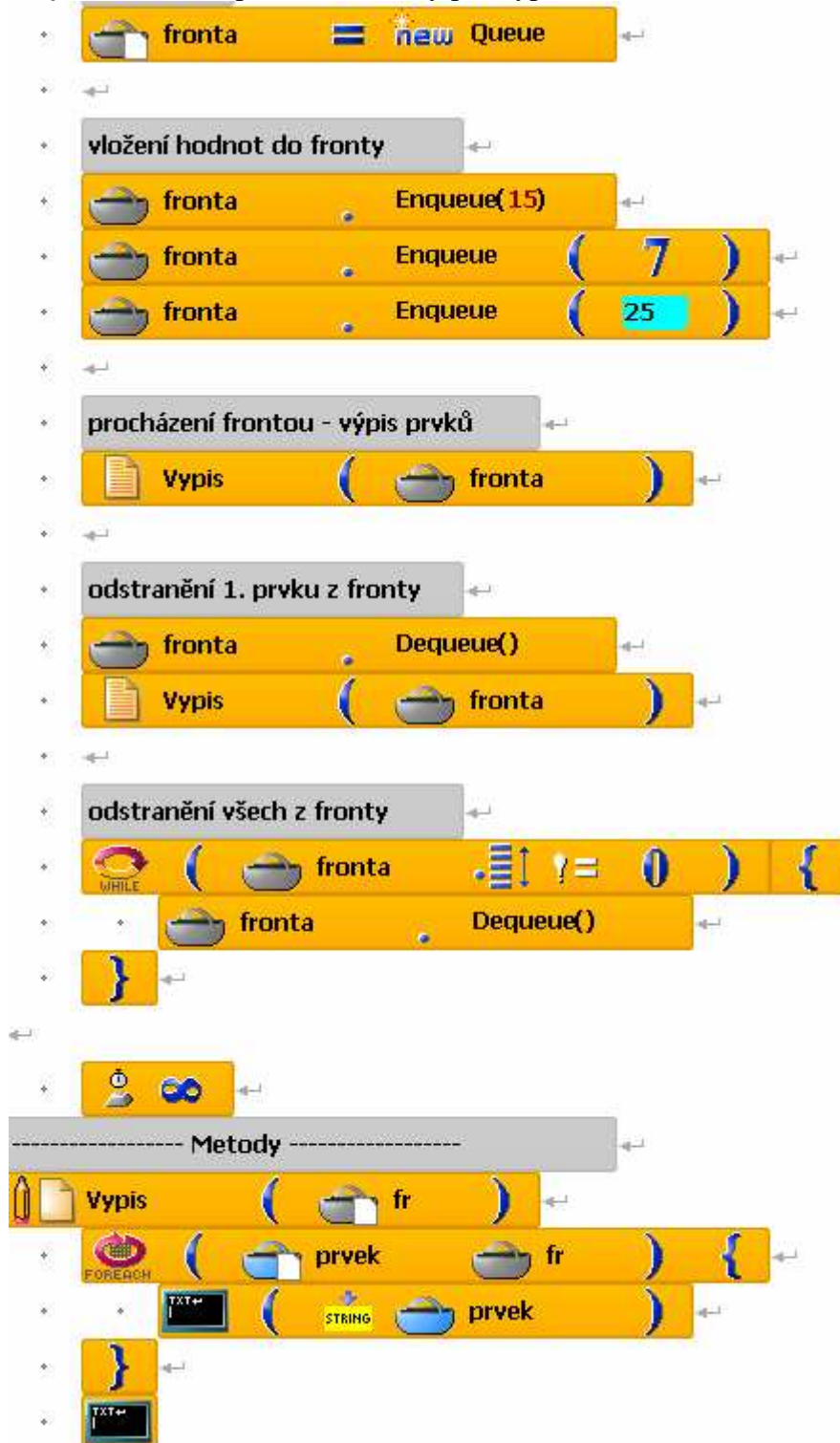
fronta.Dequeue()
foreach (prvek fronta)
    Console.WriteLine(prvek)
    
```

odstranění všech prvků

```

while (fronta.Count > 0)
    fronta.Dequeue()
    
```

7. *fronta* totéž s použitím metody pro výpis



8. **zásobník** - STACK skupina prvků, které odebíráme v pořadí od konce (od posledního prvku, který jsme vložili)

deklarace

```
zasobnik = new Stack
```

vložení hodnot do zásobníku

```
zasobnik.Push(15)
zasobnik.Push(10)
zasobnik.Push(70)
```

procházení zásobníku - výpis hodnot

```
FOREACH (prvek, zasobnik) {
    prvek.STRING
}
```

odebrání hodnoty ze zásobníku

```
zasobnik.Pop()
FOREACH (prvek, zasobnik) {
    prvek.STRING
}
```

9. **hešovací tabulka** – Hashtable – je kolekce, kde indexem je „klíč“, který nemusí být jen přirozené číslo (nejčastěji se používá řetězec“). Hodnotou je také cokoli. (libovolný objekt)
 použití: šifrování, hešování (nahrazování jedné hodnoty jinou), slovník

deklarace `vek = new Hashtable`

vložení údajů do tabulky

```
vek["tata"] = 43
vek["mama"] = 40
vek["dcera"] = 15
vek.Add("syn", 10)
```

procházení a výpis tabulky



výpis hodnoty příslušné určitému klíči



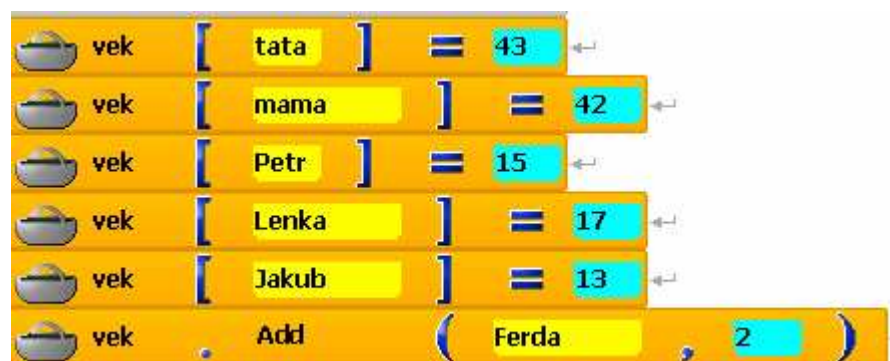
```
syn, 10
dcera, 15
mama, 40
tata, 43
```

10. *sortedlist* - řazený slovník (viz výše)

řadí celé prvky podle klíče (v našem případě vypíše prvky podle abecedy)



vložení údajů do řazeného seznamu



výpis



```
Ferda, 2
Jakub, 13
Lenka, 17
mama, 42
Petr, 15
tata, 43
```

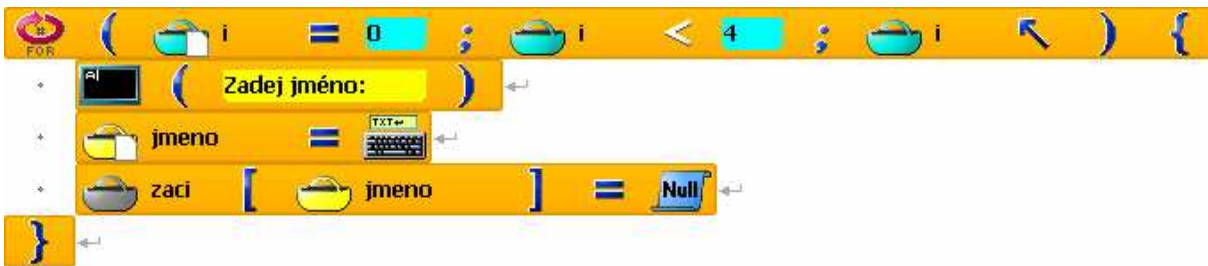
11. *sortedlist* - využití v příkladě, kde uživatel vkládá dvojici „klíč, hodnota“ (hodnota je

null – konstanta pro nedefinovanou hodnotu)

příklad seřadí podle abecedy zadaná jména



vložení údajů do řazeného seznamu



procházení tabulkou a výpis

