



# Seminář PRG

## 9. hodina - 1.11.2024

Gymnázium Voděradská 2024/2025  
Jan Borecký



# Program dneška

## Specifické typy polí

1. Listy
2. Slovníky
3. Cvičení na listy a slovníky



# Proč chceme specifická / lepší “pole”?

Nevýhody klasického pole (array):

- Pevná délka
  - Pokud bychom chtěli délku změnit, musíme vytvořit úplně nové větší pole a do něj zkopírovat všechny hodnoty, což je výpočetně náročné
- Složitě vkládání nového prvku mezi existující prvky
  - Když bychom chtěli vložit nový prvek doprostřed pole, museli bychom všechny prvky ve druhé polovině přesunout o 1 index dál
- Co když potřebujeme prvky označovat jinak, než jenom [0], [1] atd.?
  - Např. něco ve stylu students[“Alois”] = 5 (pořadí ve třídě)
- Obecně velmi málo flexibility

# Listy



# List

- Zápis: `List<datový_typ> myList = new List<datový_typ>;`
- Lze ho libovolně natahovat a zkracovat
- Pro vkládání prvků nepotřebujeme znát jeho velikost
  - Na konec listu: `myList.Add(variable);`
  - Na konkrétní index: `myList.Insert(index, variable);`
  - Více proměnných na konec listu: `myList.AddRange(variables);`
- Stejně tak pro odebírání prvků
  - `myList.Remove(element);`
  - `myList.RemoveAt(index);`
  - Při odebrání se zbytek Listu upraví podle potřeby
- Iterujeme stejně, jako s polem (for nebo foreach)



# List

- Řeší většinu neduhů, které má klasické pole (třída Array)
- Sám o sobě je striktně jednodimenzionální (narozdíl od polí)
  - Může to ale být klidně list listů (`List<List<int>>` `twoDimList`)  
čímž lze implementovat i vícedimenzionalitu
- Tím, že je “chytřejší” a flexibilnější oproti polím je List většinou paměťově náročnější a výpočetně složitější
  - Pokud nám jde o výkon, používáme pole
  - Pokud nám jde o různorodé využití a manipulaci s prvky, používáme List

# Slovníky



# Slovníky

- Zápis: `Dictionary<datový_typ_klíče, datový_typ_hodnoty> myDict = new Dictionary<datový_typ_klíče, datový_typ_hodnoty>;`
- Zápis ve slovníku se skládá z *klíče* a *hodnoty* (tzv. Key-Value Pair)
  - Klíč je "index"
  - Hodnota je... hodnota :)
  - Obojí může být libovolného datového typu
- Příklad: Slovník známek každého studenta (podle příjmení)
  - `Dictionary<string, int> grades = new Dictionary<string, int>;`
  - `grades["Novák"] = new int[] { 1, 1, 3, 2};`
  - `Console.WriteLine(grades["Novák"]);` (vypíše 1, 1, 3 a 2)





# Slovníky

- Iterace
  - Přes klíče (vlastnost dict.Keys nám vrátí pole klíčů)
    - Klasický for nebo foreach
  - Přes hodnoty (vlastnost dict.Values nám vrátí pole hodnot)
    - Klasický for nebo foreach
  - Přes dvojice klíč - hodnota
    - foreach KeyValuePair<datový\_typ\_klíče, datový\_typ\_hodnoty> nazevKVP in myDict)  
{  
  
}
- Klasický for cyklus nám tady fungovat nebude (pouze pokud bychom měli integery od 0 dál jako klíče)



# Slovníky

- Z příkladu je vidět, že opět nepotřebujeme definovat žádnou velikost
- Ve slovníku nesledujeme pouze, jestli je hodnota nějakého klíče rovna něčemu, co nás zajímá
- Zajímá nás také
  - Jestli je ve slovníku vůbec daná hodnota - `myDict.ContainsValue(value)`
  - Jestli je ve slovníku daný klíč - `myDict.ContainsKey(key)`



# Slovníky

- Slovníky využijeme, když nám klasické pole ani list nestačí
  - Většinou když potřebujeme propojit dvě specifické informace (datové typy/struktury)
- Je to (ze zmíněných) nejsložitější datová struktura, proto je práce s ní výpočetně náročnější
- Další datové struktury v C# (kterým se na semináři věnovat nebudeme, ale můžete na ně narazit)
  - Set (HashSet, SortedSet)
  - Linked List
  - Queue (a PriorityQueue)
  - Stack
  - atd.



# Cvičení

- Vytvořte si vlastní prázdný konzolový projekt a začněte pracovat na libovolném z níže zmíněných cvičení (dokončíme příští hodinu)
- 1) Vytvořte seznam (List) oblíbených jídel, ve kterém můžete libovolně přidávat a odebírat jídla  
(Kontrolujte, že nepřidává dvakrát to samé, že se nesnaží smazat jídlo, které tam není atd.)  
(Seznam po každé změně vypište.)
- 2) Vytvořte počítadlo volebních hlasů (Dictionary), do kterého přidáváte hlasy kandidátům a na konci vypíšete výsledky voleb  
(Kandidáty si určete libovolně, hlasy generujte náhodně, případně s nějakým biasem)
- 3) Vytvořte databázi cen produktů v supermarketu (Dictionary), ve kterém budete spravovat produkty a jejich ceny  
(Když aktualizujete cenu nějakého produktu, vypište, jestli se zdražil, nebo zlevnil)

# Děkuji za pozornost

**Zpětná vazba:**

<https://forms.gle/SYeWDyVITHuoYjsX7>

**Kontakt:**

Mail - [honza.borecky@seznam.cz](mailto:honza.borecky@seznam.cz)

Discord - yeenya (Yeenya#6930)

