Jazyk C# 1

4. seminář

Jakub Večeřa

Univerzita Palackého v Olomouci

18. 10. 2023

Reakce na úkoly

- Obecné vs. konkrétní řešení
- Zapouzdření
  - Jak to je s používáním getterů a setterů interně?

Reakce na úkoly

- Obecné vs. konkrétní řešení
- Zapouzdření
  - Jak to je s používáním getterů a setterů interně?
- Jednoduchost kódu:

Reakce na úkoly

- Obecné vs. konkrétní řešení
- Zapouzdření
  - Jak to je s používáním getterů a setterů interně?
- Jednoduchost kódu:

```
1 private int NonZeroAndSum(string option) {
2    int CountNonZero = 0;
3    int Sum = 0;
4    foreach (var member in matrix) {
5      if (member != 0) {
6         CountNonZero++;
7         Sum += member;
8      }
9    }
10   return option == "sum" ? Sum : CountNonZero;
11 }
```

Code convetions

- Teď už známe základy v OOP $\Rightarrow$ pojďme dát kódu jednotný tvar
- Existují konvence pojmenovávání konstruktů
- Konvence na tvar a formát
- Proč?
  - Konzistentnost – na první pohled poznáte „co je co“
  - Přenositelnost mezi kolegy
  - Automatické formátování v IDE
- `https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/coding-style/coding-conventions`
- Odteď bude vyžadováno
  - Na slidech se budu snažit dodržovat – občas poruším (úspora místa)

Modifikátory přístupu

- Není žádoucí, aby měl *kdokoliv* přístup ke všem metodám a slotům
  - Modifikátory přístupu – klíčová slova pro omezení „kdo může použít co“

Modifikátory přístupu

- Není žádoucí, aby měl *kdokoliv* přístup ke všem metodám a slotům
  - Modifikátory přístupu – klíčová slova pro omezení „kdo může použít co“

- `private` – takto označený slot/metodu může používat pouze instance dané třídy
  - Skrýváme interní funkcionalitu a interní reprezentaci

```
1 private int[,] matrix;
```

Modifikátory přístupu

- Není žádoucí, aby měl *kdokoliv* přístup ke všem metodám a slotům
  - Modifikátory přístupu – klíčová slova pro omezení „kdo může použít co“

- `private` – takto označený slot/metodu může používat pouze instance dané třídy
  - Skrýváme interní funkcionalitu a interní reprezentaci

```
1 private int[,] matrix;
```

- `public` – může používat kdokoliv

```
1 public int GetVal(int x, int y) {
2    return matrix[x, y];
3 }
```

Ukázka OOP na plátně

- Dlouhá ukázky, zdrojové kódy budou zveřejněny

Úkol

- Navrhněte třídu `IntSet` reprezentující množinu nad celými čísly s metodami:
- Pozor na duplicity prvků!
  1. `void Add(int i)` – přidání prvku i
  2. `bool Contains(int i)` – obsahuje prvek i?
  3. `void Remove(int i)` – odebrání prvku i
  4. `int Size()` – velikost množiny
  5. `IntSet Union(IntSet other)` vytvoří novou obsahující sjednocení
  6. `IntSet Intersection(IntSet other)` – vytovoří novou obsahující průnik
  7. `IntSet Subtract(IntSet other)` – vytvoří novou obsahující množ. rozdíl
  8. `bool AreEqual(IntSet other)` – obsahují množiny stejné prvky?
  9. `bool IsSubsetOf(IntSet other)` – je podmnožinou?
  10. `bool IsEmpty()` – je prázdná?