Jazyk C# 1

9. seminář

Jakub Večeřa

Univerzita Palackého v Olomouci

29. 11. 2023

Reakce na úkoly

- Některé implementace byly velmi vydařené
- Některé trpěly pouze krkolomným ovládáním
- Občas chyby v pravidlech (povinné přeskoky, doskok dámou, . . .)
- Na minulých slidech 14 dnů deadline - tedy do dnešní půlnoci
-
- Dnes proberu malá témata, která je potřeba znát, než se pustíme do GUI

Properties a Fields

- Fields == Základní datové členy tříd
  - Interní reprezentace tříd
  - Neměly by být veřejné
  - Přístupné max. přes *gettery a settery*
  - (v Javě podstatně striktnější přístup)
- Properties == Veřejné rozhraní tříd
  - Zpřístupňují fields
  - Mohou provádět validaci
  - *Gettery* různě modifikovat hodnotu
  - Případně vypočítat hodnotu z více *fieldů*
  - Mohou být automaticky generované
- Ukázka

Přetížení operátorů

- V C# je možné definovat operátory na vlastní objekty
- Lze: `+, -, !, ~ , ++, --, true, false, *, /, %, &, |, ^ , <<, >>`
- Lze párově: $==, !=, <, >, <=, >=$
- Nelze ale využívat předchozí: $+=, -=, *=, /=$, %=, &=, $|=$, $\hat{}$ =, $<<=, >>=$
- Nelze: =, ., ?:, ??, $->, =>$, f(x), as, checked, unchecked, default, delegate, is, new, sizeof, typeof

```
1  public static Complex operator +(Complex c1, Complex c2)
2  {
3  return new Complex(c1.real + c2.real, c1.imaginary +
       c2.imaginary);
4  }
```

N-tice (tuples)

- Implicitní třída pro (uspořádané n-tice) nebyla
- „Odlehčená struktura obsahující více elementů“

```
1 (int, string, double) mojeNTice = (5, "ahoj", 3.1415);
2 Console.WriteLine(mojeNTice.Item3);
```

- Pojmenování položek:

```
1 (int cislo, string retezec, double toTreti) mojeNTice = (5,
      "ahoj", 3.1415);
2 Console.WriteLine(mojeNTice.retezec);
```

  - Pojmenování dostupné pouze v čase kompilace
  - Nedostupné přes reflexi

Delegáti

- Objektové zapouzdření ukazatele na metodu (metoda `Invoke`, staticky typované)
- Mechanismus pro předávání metod jako parametr metod
- Skládání delegátů s návratovým typem `void`
  - Provedou se „po sobě“, není stejné jako skládání funkcí
- Velmi se používají v GUI
- Možno je použít pro paralelní programování (`BeginInvoke, EndInvoke` – případně samostudium)

Úkol (1/3)

- Přetižte operátory($==, !=, <, >, <=, >=, +, -, *, ++a--$) pro vaši třídu `IntMatrix` ze 3. semináře
- Vytvořte metodu `static int[] MapMathOperation(int[] array1, int[] array2, MathOperation op)`
  - Postupně aplikuje `delegate int mathOperation(int x, int y)` `MathOperation`
  - První argumenty jsou brány z `array1`, druhé z `array2`
  - Pole `array1` a `array2` jsou stejně dlouhá
  - Výsledek operace je uložen do nového pole a vrácen
- Vytvořte alespoň 2 metody splňující definici delegáta `MathOperation`
  - Metodu `MapMathOperation` s těmito metodami zavolejte

Úkol (2/3)

- Náznak volání

```
IntMatrix prvni = new IntMatrix(2, 2);
prvni.SetVal(0, 0, 2); prvni.SetVal(0, 1, 4);
prvni.SetVal(1, 0, 6); prvni.SetVal(1, 1, 8);
IntMatrix druha = new IntMatrix(2, 2);
druha.SetVal(0, 0, 1); druha.SetVal(0, 1, 3);
druha.SetVal(1, 0, 5); druha.SetVal(1, 1, 7);
IntMatrix treti = prvni + druha;
if (prvni == druha) { ... }

...

int[] arr1 = {7, 6, 1, 3};
int[] arr2 = {4, 3, 1, 2};

Console.WriteLine(MapMathOperation(arr1, arr2, someOp));
Console.WriteLine(MapMathOperation(arr1, arr2, someOp2));
```

Úkol (3/3)

- Přijde vám email s odkazem na řešení České dámy některého vašeho spolužáka
- Projděte si stažený projekt a zkuste udělit 0-8 bodů
- Napište také zdůvodnění (10 řádků textu, 12pt písmo na A4)
- Ohodnocení a zdůvodnění vložte do Upolnicku, společně s úkolem