



Programování

Práce s řetězcem a datovým typem string



Co bychom již měli znát o String?

- Nejedná se o jednoduchý datový typ
- Na string lze pohlížet jako na kolekci (pole znaků)
- Ve většině vyšších programovacích jazyků se jedná o třídu
 - Přesto pro jednoduchost mluvíme o datovém typu
- Jakožto třída má svůj atribut – Length
- S řetězcí se pojí spousta metod, které nám pomáhají při práci s řetězcí
 - Klonování, porovnávání, vyhledávání, zpracování
- String je výstupem pro čtecí funkce v případě konzolových aplikací a velké části komponent ve formulářových aplikacích

Příklady funkcí nad řetězcem

Název funkce	Co funkce dělá	Výstup funkce
Compare()	Porovnání dvou řetězců	int
Concat()	Spojení dvou a více řetězců v jeden	string
Contains()	Vyhledávání řetězce nebo znaku v řetězci	bool
EndsWith()	Zjištění zda řetězec končí konkrétním znakem nebo řetězcem	bool
StarsWith()	Zjištění zda řetězec začíná konkrétním znakem nebo řetězcem	bool
Equals()	Porovnání dvou řetězců	bool
IndexOf()	Zjištění pozice znaku podřetězce v řetězci – první výskyt	int
LastIndexOf()	Zjištění pozice posledního výskytu znaku nebo řetězce	int
Replace()	Nahrazení v řetězci	string
Split()	Rozdělení řetězce podle rozdělovače	string[]
Trim()	Ořezání řetězce o netisknutelné znaky	string

Zpracování a předzpracování řetězce

- V mnoha aplikacích potřebujeme pracovat s řetězcí
- Naše algoritmy často vyžadují specifickou podobu řetězce
 - Maximální počet znaků
 - Velikost písmen
 - Povolené znaky
 - ...
- Abychom dostali řetězce do potřebné podoby mluvíme o zpracování případně předzpracování řetězce
- Ochrana aplikace před samovolným pádem při nevalidním vstupu

Příklad – generování emailu

- Chceme vytvářet společné emaily pro zaměstnance
- Požadovaný výstup mailu je PRIJMENI.JMENO@FIRMA.CZ
- Vstupem je textové pole, kam zaměstnanec zadá své jméno a příjmení
- Možné problémy pro vytvoření validního emailu:
 - Diakritika ve jméně
 - Mezera na konci/začátku vstupu (například při kopírování)
 - Chybějící mezera mezi jménem a příjmením

Příklad – generování emailu

- Samotný algoritmus pro vytvoření mailu lze reprezentovat jednoduchou funkcí
- Vstupem jsou dva řetězce (jméno a příjmení)
- Výstupem je řetězec (nově vygenerovaný email)

```
public string generateEmail(string name, string surname)
{
    // převedení na velká písmena
    // můžeme zahrnout do předzpracování
    return $"{surname.ToUpper()}.{name.ToUpper()}@FIRMA.CZ";
}
```

- Volání této funkce předchází zpracování vstupního řetězce



Příklad – generování emailu

Předzpracování vstupu

- Lze předpokládat, že firma má více zaměstnanců
- Samotné předzpracování vstupu se hodí osamostatnit do funkce, kterou budeme opakovaně volat s každým novým vstupem
- V rámci předzpracování zvažujeme všechny možné problémy, které by mohly nastat – uživatel zadá nepěkný vstup
- Stále bychom měli mít na mysli, že existují nějaké minimální požadavky na vstup, abychom jej mohli zpracovat
- V případě, že nebudou splněny tyto minimální požadavky informujeme uživatele o této skutečnosti

Příklad – generování emailu

Předzpracování vstupu

- Možné řešení zpracování řetězce
- Předzpracování nepočítá s variantou, že má někdo dvě jména nebo příjmení
- Cílem předzpracování je vstup od uživatele dostat do vhodné podoby pro další práci s řetězcem

```
public string[] prepareInput(string inp)
{
    // oříznutí o boční mezery
    string input = inp.Trim();
    input = input.ToLower();
    // nahrazení diakritiky
    input = input.Replace("š", "s");
    input = input.Replace("č", "c");
    input = input.Replace("á", "a");
    // rozdělení vstupu podle mezery
    string[] output = input.Split(" ");
    if (output.Length != 2)
    {
        throw new Exception("Chybný formát vstupu");
    }
    // vrátíme pole {jmeno, prijmeni}
    return output;
}
```


Příklad – generování emailu

Ukázka výsledného použití

- Vstup, který bychom získali od uživatele simulujeme vytvořením nové proměnné, která by odpovídala vstupu
- Při použití proměnné **upravenyVstup** musíme myslet na to, na které pozici v poli se nachází jméno a příjmení
- Výsledný email bude mít podobu:
CERMAK.SIMON@FIRMA.CZ

```
// vstup získaný od uživatele
string vstup = " Šimon Čermák";

// příprava řetězce do potřebného formátu
string[] upravenyVstup = prepareInput(vstup);

// vytvoření emailu
string email = generateEmail(upravenyVstup[0],
                             upravenyVstup[1]);
```