

Лабораторна робота № 14.

Тема: Регулярні вирази в C#.

Мета: Ознайомитися з можливостями класу `Regex` у просторі імен `System.Text.RegularExpressions`. Навчитися створювати та застосовувати регулярні вирази для пошуку, перевірки та заміни тексту.

Зміст роботи

1. Написати регулярний вираз, який перевіряє, чи рядок містить тільки цифри.
2. Знайти всі слова, що складаються лише з великих літер.
3. Замінити всі послідовності пробілів на один пробіл.
4. Перевірити, чи рядок починається з літери і закінчується цифрою.
5. Написати регулярний вираз для перевірки, чи рядок містить рівно 5 цифр поспіль.
6. Знайти всі слова, які мають від 3 до 6 літер.
7. Замінити всі послідовності пробілів (один або більше) на одинарний пробіл.
8. Перевірити, чи рядок є поштовим індексом України (5 цифр).
9. Написати регулярний вираз, який перевіряє, чи рядок починається з великої літери.
10. Знайти всі рядки, що закінчуються знаком питання.
11. Витягнути всі слова, які стоять окремо (з використанням `\b`).
12. Перевірити, чи слово "test" зустрічається всередині іншого слова, а не окремо.
13. Створити регулярний вираз, який перевіряє, чи рядок містить слово "yes" або "no".
14. Знайти всі рядки, де зустрічається "apple" або "orange".
15. Перевірити, чи рядок закінчується словами "jpg" або "png" (корисно для перевірки формату файлів).
16. Написати вираз, який знаходить "male" або "female" у тексті анкети.

Приклад:

Знайти всі слова в тексті що починаються з літери «М»

```
// Створюємо шаблон для слів що починаються з літери "М"
string pattern = @"^\b[M]\w+";
// Створюємо екземпляр Regex
Regex rg = new Regex(pattern);

// Задаємо рядок
string authors = "Mahesh Chand, Raj Kumar, Mike Gold, Allen O'Neill,
```

```
Marshal Troll";
// Отримуємо всі співпадіння
MatchCollection matchedAuthors = rg.Matches(authors);
/// Виводимо результат
for (int count = 0; count < matchedAuthors.Count; count++)
    Console.WriteLine(matchedAuthors[count].Value);
```

Якщо необхідно ігнорувати регістр

```
// Створюємо шаблон для слова, яке починається з букви "М"
шаблон рядка = @"^\b[m]\w+";
// Створюємо екземпляр Regex
Regex rg = new Regex(pattern, RegexOptions.IgnoreCase);
```

Приклад: шаблон регулярного виразу [a-z] + та метод Regex.Split() використовуються для поділу рядка на символи без урахування їх регістру.

```
// Розбиваємо рядок за символами
string azpattern = "[a-z]+";
string str = "Asd2323b0900c1234Def5678Ghi9012Jklm";
string[] result = Regex.Split(str, azpattern,
    RegexOptions.IgnoreCase, TimeSpan.FromMilliseconds(500));
for (int i = 0; i < result.Length; i++)
{
    Console.WriteLine("{0}", result[i]);
    if (i < result.Length - 1)
        Console.Write(", ");
}
}
```

Бажано підключати бібліотеки:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
```

```
namespace RegularExpression1
{
    class Program
    {
        static void Main(string[] args)
```

Контрольні питання

1. Коли доцільно використовувати регулярні вирази?
2. Який клас подає регулярні вирази?
3. Які основні операції підтримуються класом регулярних виразів?
4. Які основні параметри конструктора класу регулярних виразів?
5. Наведіть приклад використання методу `Split()`.
6. Наведіть приклад використання методу `Matches()`.
7. Наведіть приклад використання методу `Match()`.
8. Наведіть приклад використання методу `IsMatch()`.
9. Чим відрізняються методи `Match()`, `IsMatch()` і `Matches()`?
10. Наведіть приклад використання методу `Replace()`.