MUHAMMAD SHAHZEB SHAHEEN
FA20-BCS-040
CC MID LAB
25-OCT-2023

Q1: Briefly describe the regex library of C#.

The regex library of C# is a powerful tool for working with regular expressions. It provides a set of classes and methods that allow you to match, search, and replace text based on regular expression patterns.

The most important class in the regex library is the Regex class. This class represents a regular expression and provides a variety of methods for working with it. For example, you can use the **Match()** method to find a match for the regular expression in a given string, and the Replace() method to replace all matches for the regular expression with a new string.

The regex library also includes several other classes and methods that can be useful for working with regular expressions. For example, the **MatchCollection()** class represents a collection of matches for a regular expression, and the Group class represents a group of characters within a match.

Key Classes in C# Regex Library:

1. **Regex:** The `Regex` class is the primary class in the library. It represents a compiled regular expression pattern and provides methods for pattern matching and replacement.
2. **Match:** The `Match` class represents a single match of a regular expression pattern in an input string. It provides information about the matched text and its position.
3. **MatchCollection:** This class represents a collection of `Match` objects. It is returned by methods like `Regex.Matches()` when you want to find all matches in an input string.

Basic Operations:

1. **Pattern Matching (Regex.Match):** You can use `Regex.Match()` to find the first occurrence of a regular expression pattern in an input string. It returns a `Match` object containing information about the first match.
2. **Pattern Matching (Regex.Matches**): The `Regex.Matches()` method finds all occurrences of a pattern in an input string and returns a `MatchCollection` containing all the matches.
3. **Pattern Replacement (Regex.Replace):** You can use `Regex.Replace()` to replace all occurrences of a pattern in an input string with a specified replacement string.

Common Regex Elements:

1. **Literals:** Characters that match themselves, e.g., "abc" matches the string "abc."
2. **Character Classes:** Square brackets define character classes, like `[A-Za-z]` to match any uppercase or lowercase letter.
3. **Quantifiers:** Specify how many times a character or group should appear. For example, `*` matches zero or more times, and `+` matches one or more times.
4. **Anchors:** `^` matches the start of a line, and `$` matches the end.
5. **Escape Sequences**: Backslashes `\` are used to escape special characters. For example, `\.` matches a literal period, and `\d` matches a digit.

Here is an example of how to use the regex library to match and replace text:

C#

```csharp
// Create a regular expression object
Regex regex = new Regex(@"\d+");

// Match the regular expression in the input string
Match match = regex.Match("This string contains 123 numbers");

// If there is a match, replace it with the string "numbers"
if (match.Success)
{
    string output = match.Result.Replace("123 numbers", "numbers");

    Console.WriteLine(output); // This string contains numbers
}
```

The regex library is a very powerful tool for working with text, and it can be used to solve a wide variety of problems.

Here are some of the benefits of using the regex library of C#:

- It is very efficient and can be used to process large amounts of text quickly.
- It is very flexible and can be used to create complex regular expression patterns.
- It is well-documented and there are many resources available to help you learn how to use it.

If you need to work with text in your C# applications, I highly recommend using the regex library.