Name: Muhammad Alyaan Amir

Registration No.: FA20-BCS-048

Sec: B

Answer

Regular Expressions (Regex):

Regular expressions (regex or regexp) are powerful pattern matching tools used for searching and manipulating text based on specified patterns. In C#, the `System.Text.RegularExpressions` namespace provides a library to work with regular expressions.

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:

- **Literals:** Characters that match themselves, e.g., "abc" matches the string "abc."

- **Character Classes:** Square brackets define character classes, like `[A-Za-z]` to match any uppercase or lowercase letter.

- **Quantifiers:** Specify how many times a character or group should appear. For example, `*` matches zero or more times, and `+` matches one or more times.

- **Anchors:** `^` matches the start of a line, and `$` matches the end.

- **Escape Sequences:** Backslashes `\` are used to escape special characters. For example, `\.` matches a literal period, and `\d` matches a digit.

Example Usage:

In the provided code example, we use a regex pattern to find an email address and another pattern to replace phone numbers in an input string. The `Regex.Match` and `Regex.Replace` methods are used to accomplish these tasks.

Regular expressions are a versatile tool for text processing, and understanding their syntax and usage is essential for tasks involving pattern matching and manipulation of textual data in C#.