12214994_SaurabhRana

Title

String Management System Using HashSet

Question No : 1 / 1

Console Application: String Management System Using HashSet

You are tasked with implementing a **console-based string management system** using a **HashSet** in C#.
The application should allow users to perform various operations on a collection of **unique strings**.

The program should be written within the **Program class** in the **Program.cs** file.

Constraints

- All classes must be **public**
- All methods must be **public static**
- Use **HashSet<string>** to store strings
- Duplicate strings must **not** be allowed
- Display appropriate confirmation and error messages

Operations Supported

1. Create (Add a New String)

- Prompt the user to enter a string
- Add the string to the HashSet
- If added successfully, display:
- `'<string>' has been added.`

- If the string already exists:
- `'<string>' already exists in the set.`

---

2. Read (Display All Strings)

- Display all strings currently stored in the HashSet in the format:
- `Current items in the set:`
- `<string1>`
- `<string2>`

---

3. Update (Update an Existing String)

- Prompt the user to enter the string to update
- If the string exists:
  - Prompt for a new string
  - If the new string does not already exist:
  - `'<old>' has been updated to '<new>'.`
  - If the new string already exists:
  - `Update failed. '<new>' already exists in the set.`
- If the string does not exist:
- `'<string>' does not exist in the set.`

---

4. Delete (Remove a String)

- Prompt the user to enter the string to delete
- If the string exists:
- `'<string>' has been removed.`
- If the string does not exist:
- `'<string>' does not exist in the set.`

---

5. Exit

- Terminate the program

Instructions

Display the following menu repeatedly until the user selects Exit:

```
Choose an operation:
1: Create (Add a new string)
2: Read (Display all strings)
```

```
3: Update (Update an existing string)
4: Delete (Remove a string)
5: Exit
```

Prompt the user to enter a choice between **1–5**.

Input Format

- Menu choice (integer 1–5)
- Strings entered based on selected operation

Output Format

- Confirmation messages for create, update, delete
- Display list for read operation
- Error messages when strings are not found or duplicates exist

---

Sample Flow (As Per Screenshots)

Create – Add String

```
Enter your choice: 1
Enter the string to add: Hello
'Hello' has been added.
```

Duplicate Entry

```
Enter the string to add: Hello
'Hello' already exists in the set.
```

Read – Display Strings

```
Enter your choice: 2
Current items in the set:
Hello
World!
```

Update – Existing String

```
Enter your choice: 3
Enter the string to update: World!
Enter the new string: Virat
'World!' has been updated to 'Virat'.
```

String Not Found

```
Enter the string to update: Humans
'Humans' does not exist in the set.
```

Delete – Remove String

```
Enter your choice: 4
Enter the string to delete: Hello
'Hello' has been removed.
```

Delete Not Found

```
Enter the string to delete: Me
'Me' does not exist in the set.
```

Exit

```
Enter your choice: 5
Exiting...
```

C# Solution (Program.cs)

```
public class Program

{

    public static void AddString(HashSet<string> set)

    {

        Console.Write("Enter the string to add: ");

        string value = Console.ReadLine();
        if (string.IsNullOrWhiteSpace(value))
        {
            Console.WriteLine("Invalid input.");
            return;
        }

        if (set.Add(value))
        {
            Console.WriteLine($"'{value}' has been added.");
        }
        else
        {
            Console.WriteLine($"'{value}' already exists in the set.");
        }
    }

    public static void DisplayStrings(HashSet<string> set)
    {
        Console.WriteLine("Current items in the set:");

        if (set.Count == 0)
        {
            Console.WriteLine("Nothing in the set.");
            return;
        }
        foreach (string item in set)
        {
            Console.WriteLine(item);
        }
    }

    public static void UpdateString(HashSet<string> set)
    {
        Console.Write("Enter the string to update: ");
        string oldValue = Console.ReadLine();

        if (!set.Contains(oldValue))
        {
            Console.WriteLine($"'{oldValue}' does not exist in the set.");
            return;
        }

        Console.Write("Enter the new string: ");
        string newValue = Console.ReadLine();

        if (string.IsNullOrWhiteSpace(newValue))
        {
            Console.WriteLine("Invalid input.");
            return;
        }

        if (set.Contains(newValue))
        {
            Console.WriteLine($"Update failed. '{newValue}' already exists in the set.");
            return;
        }

        set.Remove(oldValue);
        set.Add(newValue);

        Console.WriteLine($"'{oldValue}' has been updated to '{newValue}'.");
    }

    public static void DeleteString(HashSet<string> set)
    {
        Console.Write("Enter the string to delete: ");
        string value = Console.ReadLine();

        if (set.Remove(value))
        {
            Console.WriteLine($"'{value}' has been removed.");
        }
        else
        {
            Console.WriteLine($"'{value}' does not exist in the set.");
        }
    }
    public static void Main()
    {
        HashSet<string> stringSet = new HashSet<string>();

        while (true)
        {
            Console.WriteLine("Choose an operation:");
            Console.WriteLine("1: Create (Add a new string)");
            Console.WriteLine("2: Read (Display all strings)");
            Console.WriteLine("3: Update (Update an existing string)");
            Console.WriteLine("4: Delete (Remove a string)");
            Console.WriteLine("5: Exit");

            Console.Write("Enter your choice: ");
            string input = Console.ReadLine();

            if (!int.TryParse(input, out int choice))
            {
                Console.WriteLine("Invalid choice.");
                continue;
            }
```

```
switch (choice)
{
    case 1:
        AddString(stringSet);
        break;

    case 2:
        DisplayStrings(stringSet);
        break;

    case 3:
        UpdateString(stringSet);
        break;

    case 4:
        DeleteString(stringSet);
        break;

    case 5:
        Console.WriteLine("Exiting...");
        return;

    default:
        Console.WriteLine("Invalid choice.");
        break;
```

```
            }
        }
    }
}
```