

Generics and Interfaces with a Repository Pattern - Task 8

Program that demonstrates a generic in-memory repository pattern with CRUD operations using a console UI for managing Student records :

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

namespace GenericRepositoryDemo
{
    public interface IRepository<T> where T : BaseEntity
    {
        void Add(T item);
        T Get(int id);
        IEnumerable<T> GetAll();
        void Update(T item);
        void Delete(int id);
    }

    public abstract class BaseEntity
    {
        public int Id { get; set; }
    }

    public class Student : BaseEntity
    {
        public string Name { get; set; }
        public int Age { get; set; }
    }

    public class InMemoryRepository<T> : IRepository<T> where T : BaseEntity
    {
        private readonly List<T> _items = new List<T>();
        private int _nextId = 1;

        public void Add(T item)
        {
            item.Id = _nextId++;
            _items.Add(item);
        }

        public T Get(int id) => _items.FirstOrDefault(x => x.Id == id);

        public IEnumerable<T> GetAll() => _items;
    }
}
```

```
public void Update(T item)
{
    var existing = Get(item.Id);
    if (existing != null)
    {
        int index = _items.IndexOf(existing);
        _items[index] = item;
    }
}

public void Delete(int id)
{
    var item = Get(id);
    if (item != null)
        _items.Remove(item);
}
}

// Main Program
class Program
{
    static void Main()
    {
        IRepository<Student> studentRepo = new InMemoryRepository<Student>();

        while (true)
        {
            Console.WriteLine("\n--- Student Repository Menu ---");
            Console.WriteLine("1. Add Student");
            Console.WriteLine("2. View All Students");
            Console.WriteLine("3. Update Student");
            Console.WriteLine("4. Delete Student");
            Console.WriteLine("5. Exit");
            Console.Write("Enter your choice: ");

            switch (Console.ReadLine())
            {
                case "1":
                    Console.Write("Enter Name: ");
                    var name = Console.ReadLine();
                    Console.Write("Enter Age: ");
                    int age = int.Parse(Console.ReadLine());

                    studentRepo.Add(new Student { Name = name, Age = age });
                    Console.WriteLine("Student added successfully!");
                    break;
            }
        }
    }
}
```

```
        case "2":
            Console.WriteLine("\nList of Students:");
            foreach (var student in studentRepo.GetAll())
            {
                Console.WriteLine($"ID: {student.Id}, Name: {student.Name}, Age:
{student.Age}");
            }
            break;

        case "3":
            Console.Write("Enter ID to update: ");
            int idToUpdate = int.Parse(Console.ReadLine());
            var studentToUpdate = studentRepo.Get(idToUpdate);
            if (studentToUpdate != null)
            {
                Console.Write("Enter new Name: ");
                studentToUpdate.Name = Console.ReadLine();
                Console.Write("Enter new Age: ");
                studentToUpdate.Age = int.Parse(Console.ReadLine());
                studentRepo.Update(studentToUpdate);
                Console.WriteLine("Student updated.");
            }
            else
                Console.WriteLine("Student not found.");
            break;

        case "4":
            Console.Write("Enter ID to delete: ");
            int idToDelete = int.Parse(Console.ReadLine());
            studentRepo.Delete(idToDelete);
            Console.WriteLine("Student deleted.");
            break;

        case "5":
            return;

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

Output :

Adding a student :

```
--- Student Repository Menu ---  
1. Add Student  
2. View All Students  
3. Update Student  
4. Delete Student  
5. Exit  
Enter your choice:  
1
```

```
Enter Name:  
→ karu  
  
Enter Age:  
→ 20  
  
Student added successfully!
```

Viewing all students:

```
--- Student Repository Menu ---  
1. Add Student  
2. View All Students  
3. Update Student  
4. Delete Student  
5. Exit  
Enter your choice:  
→ 2
```

```
List of Students:  
ID: 1, Name: karu, Age: 20  
ID: 2, Name: harini, Age: 20
```

Updating student details:

```
Enter your choice:
```

```
3
```

```
Enter ID to update:
```

```
1
```

```
Enter new Name:
```

```
karunya
```

```
Enter new Age:
```

```
21
```

```
Student updated.
```

```
Enter your choice:
```

```
2
```

```
List of Students:
```

```
ID: 1, Name: karunya, Age: 21
```

```
ID: 2, Name: harini, Age: 20
```

Deleting student:

```
3. Update Student
```

```
4. Delete Student
```

```
5. Exit
```

```
Enter your choice:
```

```
4
```

```
Enter ID to delete:
```

```
2
```

```
Student deleted.
```

```
Enter your choice:
```

```
→ 2
```

```
List of Students:
```

```
ID: 1, Name: karunya, Age: 21
```