**Лабораторная работа №9**

**Коллекции, классы-прототипы, обобщения**

**Задание 1**

**Реализация системы отмены действий (Undo/Redo) с помощью Stack**

* + Класс **Command** представляет действие с описанием.
  + Класс **CommandManager** управляет двумя стеками: выполненных команд и отмененных команд.
  + Реализовать методы Execute(Command), Undo(), Redo().

**Код**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Task1

{

public class CommandManager

{

private Stack<Command> executedCommands = new Stack<Command>();

private Stack<Command> undoneCommands = new Stack<Command>();

public void Execute(Command command)

{

command.Execute();

executedCommands.Push(command);

undoneCommands.Clear();

}

public void Undo()

{

if (executedCommands.Count > 0)

{

Command command = executedCommands.Pop();

command.Undo();

undoneCommands.Push(command);

}

else

{

Console.WriteLine("Никаких команд для отмены.");

}

}

public void Redo()

{

if (undoneCommands.Count > 0)

{

Command command = undoneCommands.Pop();

command.Execute();

executedCommands.Push(command);

}

else

{

Console.WriteLine("Никаких команд для повтора.");

}

}

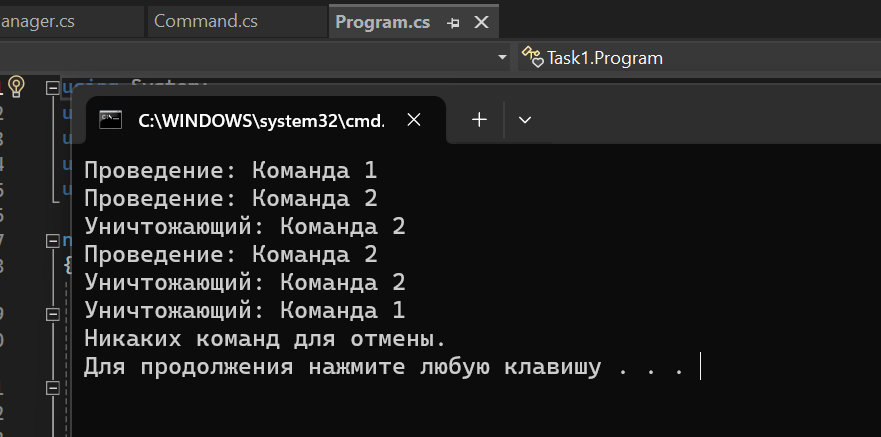
}

}

**Таблица 9.1 – Входные и выходные данные**

|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
|  | **Проведение: Команда 1**  **Проведение: Команда 2**  **Уничтожающий: Команда 2**  **Проведение: Команда 2**  **Уничтожающий: Команда 2**  **Уничтожающий: Команда 1**  **Нет команд для отмены.** |

**Анализ результата**

****

**Задание 2**

**Реализация собственной коллекции MyList<T>**

* + **Класс MyList<T>** – аналог List<T>.
  + Должен поддерживать методы Add(T item), Remove(T item), Find(Predicate<T> predicate), Sort(Comparison<T> comparison).
  + **Класс ListManager<T>** – управляет списком и выполняет сортировку, поиск по критериям.

**Код**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Task2

{

internal class ListManager<T>

{

private MyList<T> myList;

public ListManager()

{

myList = new MyList<T>();

}

public void AddItem(T item)

{

myList.Add(item);

}

public void RemoveItem(T item)

{

if (!myList.Remove(item))

{

Console.WriteLine("Предмет не найден.");

}

}

public T FindItem(Predicate<T> predicate)

{

return myList.Find(predicate);

}

public void SortItems(Comparison<T> comparison)

{

myList.Sort(comparison);

}

public void DisplayItems()

{

for (int i = 0; i < myList.Count; i++)

{

Console.WriteLine(myList[i]);

}

}

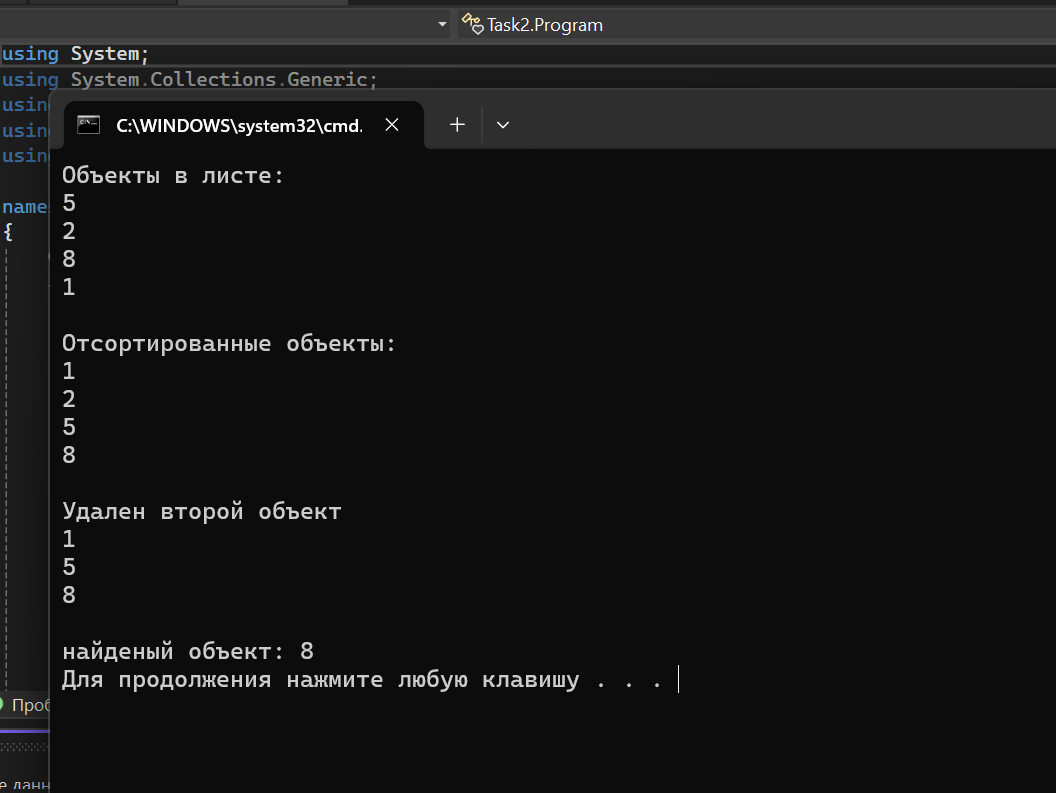
}

}

**Таблица 9.2 – Входные и выходные данные**

|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
|  | **Объекты в листе:**  **5**  **2**  **8**  **1**  **Отсортированные объекты:**  **1**  **2**  **5**  **8**  **Удален второй объект**  **1**  **5**  **8**  **Найденный объект: 8** |

**Анализ результата**

****