**18** **Коллекции. Классы-прототипы**

Задание 1. Пусть символ # определен в текстовом редакторе как стирающий символ Backspace, т.е. строка abc#d##c в действительности является строкой ac. Дан текст, в котором встречается символ #. Преобразовать его с учетом действия этого символа.

Листинг программы:

namespace Space

{

class Program

{

static void Main(string[] args)

{

Console.Write("Введите строку с символом #: ");

string input = Console.ReadLine();

string result = "";

foreach (char c in input)

{

if (c != '#')

{

result += c;

}

else if (result.Length > 0)

{

result = result.Remove(result.Length - 1);

}

}

Console.WriteLine("Преобразованная строка: " + result);

}

}

}

Таблица 18.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Привет, я#купил# помидоры#. | Привет, купи помидор. |

Анализ результатов:

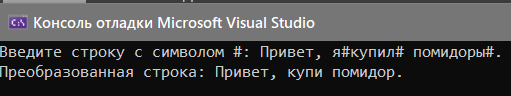


Рисунок 18.1 – Результат работы программы

Задание 2. Создать класс-прототип, с обобщенным методом, с методами добавления, удаления, клонирования экземпляров класса.

В основной программе продемонстрировать:

функциональность созданного класса и всех его методов;

использование обнуляемых типов,

использование абстрактных структур (List<P> Dictionary<Key,Value&gt;).

Листинг программы:

namespace Space

{

abstract class Sport

{

protected List<string> players;

public Sport()

{

players = new List<string>();

}

public abstract void Play();

public void AddPlayer(string player)

{

players.Add(player);

}

public void RemovePlayer(string player)

{

players.Remove(player);

}

public Sport Clone()

{

return (Sport)this.MemberwiseClone();

}

public void PrintPlayers()

{

Console.WriteLine();

Console.Write("Список игроков: ");

foreach (string player in players)

{

Console.Write(player + " ");

}

Console.WriteLine();

}

public void AddPlayers(int a)

{

int i = 0;

while (i < a)

{

Console.Write("Введите имя игрока: ");

string playerName = Console.ReadLine();

AddPlayer(playerName);

i++;

}

}

public void RemovePlayers(int b)

{

int i = 0;

while (i < b)

{

Console.Write("Введите имя игрока, которого нужно выгнать: ");

string playerName = Console.ReadLine();

RemovePlayer(playerName);

i++;

}

}

}

class Football : Sport

{

public override void Play()

{

Console.WriteLine();

Console.WriteLine("Играем в футбол с {0} игроками", players.Count);

Console.WriteLine();

}

}

class Program

{

static void Main(string[] args)

{

Console.Write("Сколько человек играет: ");

int a = Convert.ToInt32(Console.ReadLine());

Football football = new Football();

football.AddPlayers(a);

football.PrintPlayers();

football.Play();

Console.Write("Сколько человек выгоняем: ");

int b = Convert.ToInt32(Console.ReadLine());

Football football2 = football.Clone() as Football;

football2.RemovePlayers(b);

football2.PrintPlayers();

football2.Play();

}

}

}

Таблица 18.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 6  Игнат  Рувим  Миша  Стас  Рустем  Толя  2 | Игнат Рувим Миша Стас Рустем Толя  Игнат Рувим Стас Рустем |

Анализ результатов:

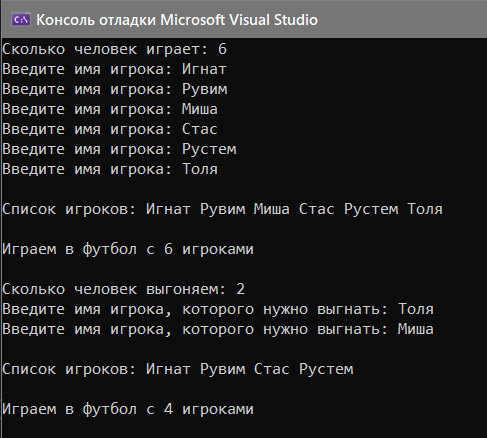


Рисунок 18.2 – Результат работы программы