

## 18 КОЛЛЕКЦИИ. КЛАССЫ-ПРОТОТИПЫ

Задание 1. Решить задачу с использованием класса Stack.

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

Листинг программы:

```
try
{
    Console.WriteLine("Введите строку: ");
    string str = Console.ReadLine();
    var textStack = new Stack<char>();
    for (int i = 0; i < str.Length; i++)
        textStack.Push(str[i]);
    Console.WriteLine(EraseCharacters(textStack));
}
catch (Exception ex)
{
    Console.WriteLine(ex.Message);
}

string EraseCharacters(Stack<char> text)
{
    int counter = 0;
    string str = string.Empty;

    while (text.Count > 0)
    {
        char c = text.Pop();
        if (c == '#')
            counter++;
        else if (counter > 0)
            counter--;
        else str += c;
    }
}
```

					УП 2-40 01 01.31 ТП.2471.22.18		
Изм.	Лист	№ докум	Подп.	Дата			
Разраб.		Мушинский М.С.			КОЛЛЕКЦИИ. КЛАССЫ-ПРОТОТИПЫ		
Пров.		Толочко П.С.					
Н.контр.							
Утв.							
						Лит	Лист
							Листов
						Гродненский ГКТТид	

```

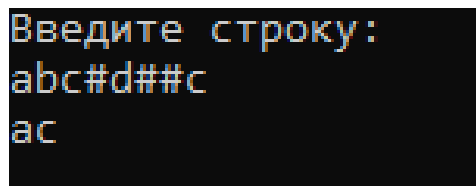
    }
    char[] chars = new char[str.Length];
    chars = str.Select(x => x).ToArray();
    Array.Reverse(chars);
    str = new string(chars);
    return str;
}

```

Таблица 18.1 – Входные и выходные данные

Входные данные	Выходные данные
abc#d#c	ac

Анализ результатов:



```

Введите строку:
abc#d#c
ac

```

Рисунок 18.1 – Код разработанной библиотеки  
Источник: собственная разработка

Задание 2. Решить следующую задачу с использованием класса Queue.

Дан файл, содержащий числа. За один просмотр файла напечатать элементы файла в следующем порядке: сначала все числа, из интервала [a,b], потом все числа, меньшие a, потом все числа, большие b, сохраняя исходный порядок в каждой группе чисел.

Листинг программы:

```

int a = 2, b = 6;
Queue<int> fromAToB = new Queue<int>();
Queue<int> lessThenA = new Queue<int>();
Queue<int> moreThenB = new Queue<int>();
int number;
using (StreamReader reader = new StreamReader(@"file.txt", Encoding.UTF8))
{
    while (!reader.EndOfStream)
    {
        number = int.Parse(reader.ReadLine());
        if (number < a)
            lessThenA.Enqueue(number);
    }
}

```

```

else if (number > b)

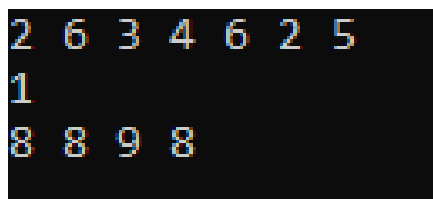
    moreThenB.Enqueue(number);
else
    fromAToB.Enqueue(number);
}
}
Console.WriteLine(string.Join(" ", fromAToB));
Console.WriteLine(string.Join(" ", lessThenA));
Console.WriteLine(string.Join(" ", moreThenB));

```

Таблица 18.2 – Входные и выходные данные

Входные данные	Выходные данные
file.txt	2, 6, 3, 4, 6, 2, 5 1 8, 8, 9, 8

Анализ результатов:



```

2 6 3 4 6 2 5
1
8 8 9 8

```

Рисунок 18.2 – Результат работы программы  
Источник: собственная разработка