

Министерство транспорта Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«Российский университет транспорта»
(ФГАОУ ВО РУТ(МИИТ), РУТ (МИИТ))

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

Лабораторная работа № 11

по дисциплине: «Программирование и основы алгоритмизации»
на тему: «Методы»

Выполнил: ст. гр.

Денг Денг Гак Луал

Вариант №13

27.05.2025

Проверил: к.т.н., доц. Сафронов А.И

.....

Москва – 2025 г.

1. Цель

разработка консольного приложения на языке Visual C# в среде Microsoft Visual Studio для решения задач по теме "Методы", с соблюдением классификации методов (типизированные/нетипизированные) и параметров (возвращаемые по ссылке/инициализируемые методом), согласно индивидуальному заданию.

2. Формулировка задачи

Разработать нетипизированный метод с префиксом "_m" для шифрования и дешифрования текстовых сообщений. Метод должен принимать строку по ссылке и логический параметр, определяющий направление операции. Алгоритм шифрования заключается в обмене каждым вторым символом с начала строки с каждым вторым символом с конца строки. Реализовать контроль ввода данных и обеспечить корректную обработку различных типов строк.

3. Блок-схема алгоритма

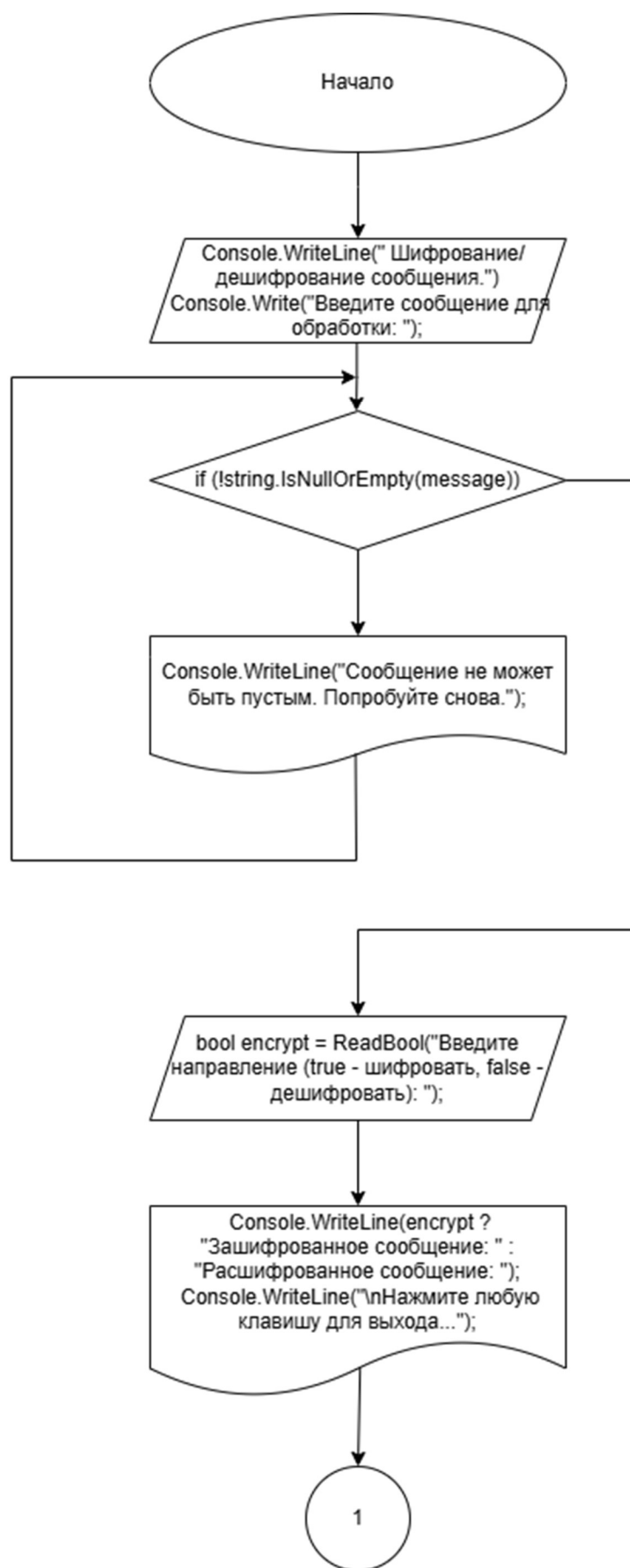


Рисунок 3.1 – Блок-схема алгоритма

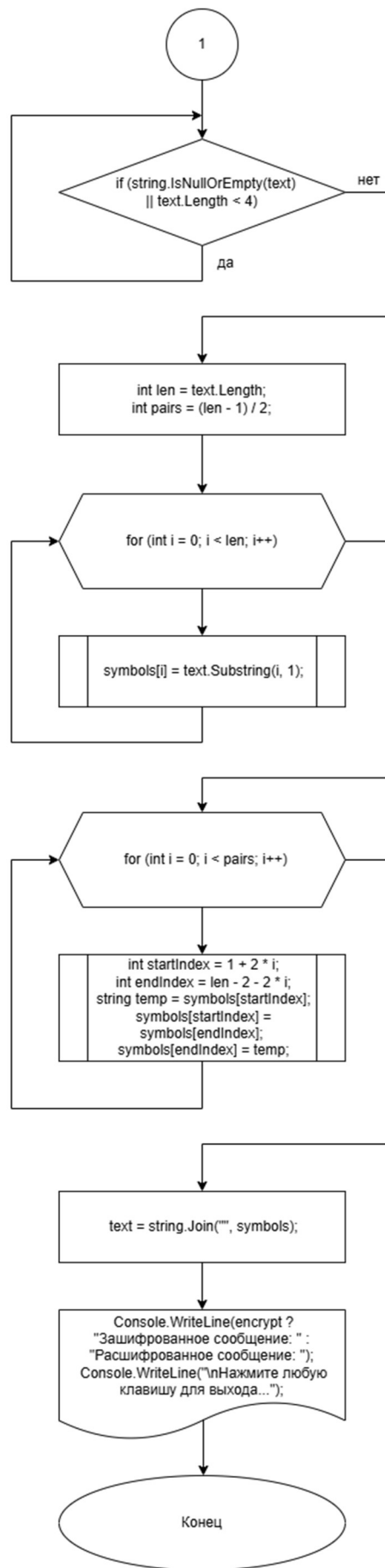


Рисунок 3.2 – Блок-схема алгоритма

4. Подбор тестовых примеров

Тест 1:

- Вводим: "abcdef", true
- Ожидаемый результат: " aedcbf"

Тест 2:

- Вводим: "afcdbe", false
- Ожидаемый результат: " abdcfe"

Тест 3: Четное количество символов

- Вводим: "12345678", true
- Ожидаемый результат: " 17654328"

5. Листинг (код) программы

```
using System;
```

```
class Program
```

```
{
```

```
    static void Main()
```

```
    {
```

```
        Console.WriteLine(" Шифрование/дешифрование сообщения.");
```

```
        // Ввод сообщения с проверкой на пустоту
```

```
        string message;
```

```
        while (true)
```

```
        {
```

```
            Console.Write("Введите сообщение для обработки: ");
```

```
            message = Console.ReadLine();
```

```
            if (!string.IsNullOrEmpty(message))
```

```
                break;
```

```
            Console.WriteLine("Сообщение не может быть пустым. Попробуйте снова.");
```

```
        }
```

```
// Ввод направления шифрования с контролем
bool encrypt = ReadBool("Введите направление (true - шифровать, false - дешифровать): ");

// Вызов метода шифрования/дешифрования
_m(ref message, encrypt);

Console.WriteLine(encrypt ? "Зашифрованное сообщение: " : "Расшифрованное сообщение: ");
Console.WriteLine(message);

Console.WriteLine("\nНажмите любую клавишу для выхода...");
Console.ReadKey();
}
```

```
/// <summary>
/// Нетипизированный (void) метод шифрования/дешифрования.
/// Каждый второй символ с начала меняется местами с каждым вторым символом с конца.
/// </summary>
/// <param name="text">Строка передаётся по ссылке и изменяется</param>
/// <param name="encrypt">true - шифровать, false - дешифровать</param>
static void _m(ref string text, bool encrypt)
{
    if (string.IsNullOrEmpty(text) || text.Length < 4)
        return;

    int len = text.Length;
    int pairs = (len - 1) / 2;

    // Разбиваем строку на массив строк длиной 1 (символы в виде строк)
    string[] symbols = new string[len];
    for (int i = 0; i < len; i++)
    {
        symbols[i] = text.Substring(i, 1);
    }
}
```

```

    }

    // Меняем местами каждый второй символ с начала и с конца
    for (int i = 0; i < pairs; i++)
    {
        int startIndex = 1 + 2 * i;
        int endIndex = len - 2 - 2 * i;

        string temp = symbols[startIndex];
        symbols[startIndex] = symbols[endIndex];
        symbols[endIndex] = temp;
    }

    // Собираем обратно строку
    text = string.Join("", symbols);
}

/// <summary>
/// Метод для безопасного чтения bool с подсказкой, с повторным вводом при ошибке
/// </summary>
/// <param name="prompt">Сообщение-подсказка</param>
/// <returns>Введённое значение bool</returns>
static bool ReadBool(string prompt)
{
    bool result;
    while (true)
    {
        Console.Write(prompt);
        string input = Console.ReadLine();
        if (bool.TryParse(input, out result))
        {
            return result;
        }
    }
}

```

```

    }
else
{
    Console.WriteLine("Ошибка ввода! Пожалуйста, введите 'true' или 'false'.");
}
}
}
}
}

```

6. Расчет тестовых примеров на ПК

```

Шифрование/дешифрование сообщения.
Введите сообщение для обработки: abcdef
Введите направление (true – шифровать, false – дешифровать): true
Зашифрованное сообщение:
aedcbf

Нажмите любую клавишу для выхода...
|

```

Рисунок 6.1 – Тестовый пример №1

```

Шифрование/дешифрование сообщения.
Введите сообщение для обработки: afcdbe
Введите направление (true – шифровать, false – дешифровать): false
Расшифрованное сообщение:
abdcfe

Нажмите любую клавишу для выхода...

```

Рисунок 6.2 – Тестовый пример №2

```

Шифрование/дешифрование сообщения.
Введите сообщение для обработки: 12345678
Введите направление (true – шифровать, false – дешифровать): true
Зашифрованное сообщение:
17654328

Нажмите любую клавишу для выхода...
|

```

Рисунок 6.3– Тестовый пример №3

7. Вывод

В ходе работы создана программа на C# для шифрования сообщений путём обмена каждым вторым символом с начала и конца строки. Реализован нетипизированный метод с передачей строки по ссылке, контроль ввода данных и обработка различных типов строк. Программа соответствует всем требованиям и демонстрирует стабильную работу.