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 № 2

по дисциплине: «Программирование и основы алгоритмизации»

на тему: «Типовая задача ввода-вывода. Побуждающие сообщения. Присвоение»

Выполнил: ст. гр. ТУУ-111

Попов В.А.

Вариант №3

26.12.2024

(дата выполнения)

Проверил: к.т.н., доц. Сафронов А.И.

\_\_\_\_\_\_\_\_\_\_\_

(дата приёмки)

Москва – 2024 г.

1. **Цель**

Целью данной работы является освоение различных методов ввода-вывода информации в консоль с использованием интегрированной среды разработки Microsoft Visual Studio и языка программирования C#. В работе необходимо реализовать вывод сообщений с использованием строковых констант, интерполяции строк, конкатенации строк и строковых методов.

1. **Формулировка задачи**
   1. **Параметрический вывод (Константы)**

Формулировка задачи:

Организовать вывод сообщения в консоль с использованием параметрического способа, где каждый элемент сообщения подставляется как параметр. Сообщение должно быть задано строковой константой вида:

"{0} голова {1}, {2} {3} – лучше", а значения параметров заданы константами:

I = 1

II = "хорошо"

III = 'a'

IV = 2

* 1. **Интерполяция строк (Переменные)**

Формулировка задачи:

Используя механизм интерполяции строк C#, организовать вывод сообщения в консоль, формирующего предложение из заданных переменных. Каждая переменная обозначает часть строки и формирует осмысленную фразу, записанную кириллицей.

Переменные:

GUP = "Руки"

Mos = "по"

Gor = "бо"

Trans = "рют"

predlagaet = 1

voditelyam = ','

khorosho = "знание"

oplachivaemuyu = '-'

rabotu = 1000

Фраза: $"{GUP} {Mos}{Gor}{Trans} {predlagaet}{voditelyam} {khorosho} {oplachivaemuyu} {rabotu}".

* 1. **Конкатенация строк**

Формулировка задачи:

Составить осмысленное предложение, используя конкатенацию строк и символов. Каждое слово выводимой фразы должно быть представлено отдельной символьной константой или переменной, а символы объединены с использованием оператора +.

Фраза: "Бедность – не порок, а большое несчастье."

* 1. **Строковые методы**

Формулировка задачи:

Использовать стандартные строковые методы C# для обработки текста. Удалить опечатки в словах через метод Remove().

Исходная строка:

"Веръшины, пятынистые, сеедой, порялдели, очищають, трясеёт, приппекает, помешарть"

1. **Схема алгоритма**
   1. **Параметрический вывод**

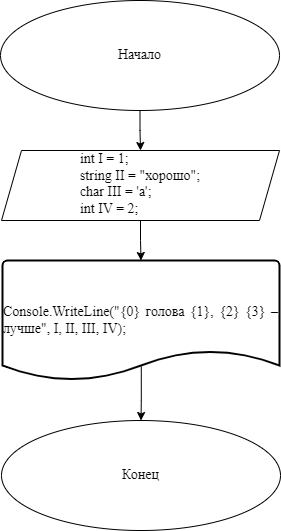
****

Рисунок 3.1.1 – Схема алгоритма параметрического вывода

* 1. **Интерполяция строк**

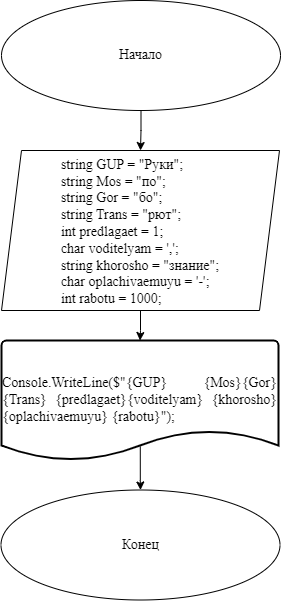
****

Рисунок 3.2.1 – Схема алгоритма интерполяции строк

* 1. **Конкатенация строк**

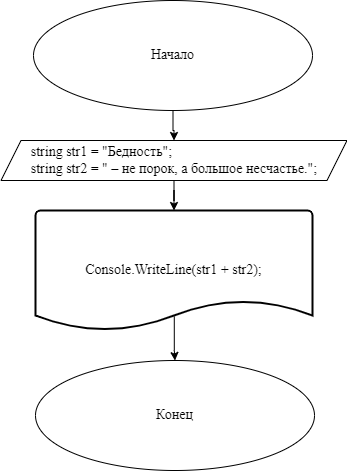
****

Рисунок 3.3.1 – Схема алгоритма конкатенации строк

* 1. **Строковые методы**

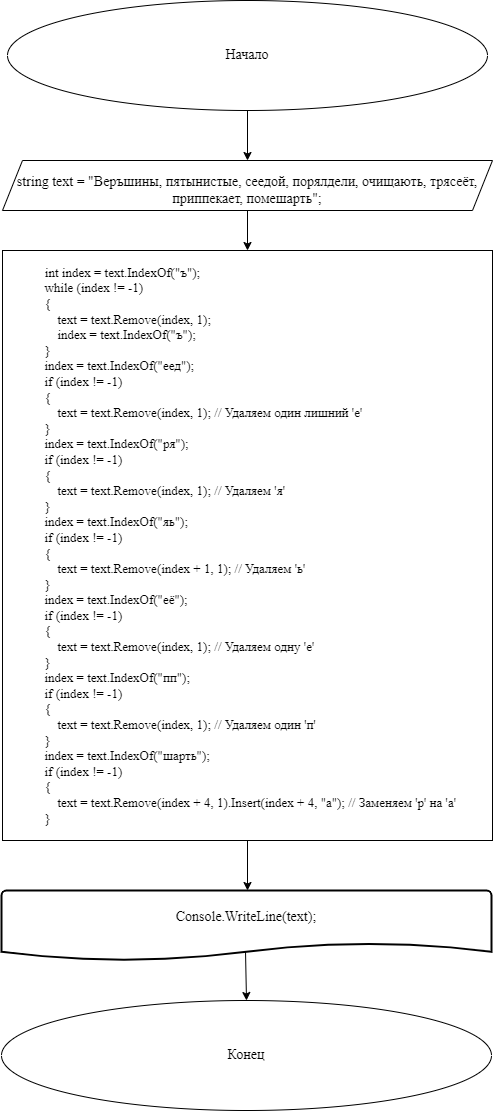
****

Рисунок 3.4.1 – Схема алгоритма строковых методов

1. **Подбор тестовых примеров**
   1. **Параметрический вывод:**

Исходная строка: "{0} {1} {4}мерь{2}{3} {1} {4}режь"

Подставляемые значения:

sl0 = 7

sl1 = "раз"

sl2 = ", "

sl3 = 1

sl4 = "от"

Ожидаемый результат: 7 раз отмерь, 1 раз отрежь

* 1. **Интерполяция строк:**

Исходные данные:

GUP = "Руки"

Mos = "по"

Gor = "бо"

Trans = "рют"

predlagaet = 1

voditelyam = ','

khorosho = "знание"

oplachivaemuyu = '-'

rabotu = 1000

Ожидаемый результат: Руки по борют 1, знание – 1000

* 1. **Конкатенация строк:**

Исходные данные:

Строка: "Бедность – не порок, а большое несчастье."

Ожидаемый результат: Бедность – не порок, а большое несчастье.

* 1. **Строковые методы:**

Исходная строка: "Веръшины, пятынистые, сеедой, порялдели, очищають, трясеёт, приппекает, помешарть"

Ожидаемый результат: "Вершины, пятнистые, седой, поредели, очищают, трясет, припекает, помешать"

1. **Листинг (Код программы)**
   1. **Параметрический вывод:**

using System;

class Program

{

static void Main()

{

// Объявление и инициализация констант

int I = 1;

string II = "хорошо";

char III = 'a';

int IV = 2;

// Форматированный вывод сообщения

Console.WriteLine("{0} голова {1}, {2} {3} – лучше", I, II, III, IV);

}

}

* 1. **Интерполяция строк:**

using System;

class Program

{

static void Main()

{

// Объявление переменных

string GUP = "Руки";

string Mos = "по";

string Gor = "бо";

string Trans = "рют";

int predlagaet = 1;

char voditelyam = ',';

string khorosho = "знание";

char oplachivaemuyu = '-';

int rabotu = 1000;

// Формирование строки с использованием интерполяции

Console.WriteLine($"{GUP} {Mos}{Gor}{Trans} {predlagaet}{voditelyam} {khorosho} {oplachivaemuyu} {rabotu}");

}

}

* 1. **Конкатенация строк:**

using System;

class Program

{

static void Main()

{

// Объявление переменных

string str1 = "Бедность";

string str2 = " – не порок, а большое несчастье.";

// Конкатенация строк

Console.WriteLine(str1 + str2);

}

}

* 1. **Строковые методы:**

using System;

class Program

{

static void Main()

{

// Исходная строка с ошибками

string text = "Веръшины, пятынистые, сеедой, порялдели, очищають, трясеёт, приппекает, помешарть";

// Исправление текста с использованием Remove() и вставки исправлений

// Удаление 'ъ'

int index = text.IndexOf("ъ");

while (index != -1)

{

text = text.Remove(index, 1);

index = text.IndexOf("ъ");

}

// Исправление "еедой" -> "едой"

index = text.IndexOf("еед");

if (index != -1)

{

text = text.Remove(index, 1); // Удаляем один лишний 'е'

}

// Исправление "порялдели" -> "пордели"

index = text.IndexOf("ря");

if (index != -1)

{

text = text.Remove(index, 1); // Удаляем 'я'

}

// Исправление "очищають" -> "очищают"

index = text.IndexOf("яь");

if (index != -1)

{

text = text.Remove(index + 1, 1); // Удаляем 'ь'

}

// Исправление "трясёёт" -> "трясёт"

index = text.IndexOf("её");

if (index != -1)

{

text = text.Remove(index, 1); // Удаляем одну 'е'

}

// Исправление "приппекает" -> "припекает"

index = text.IndexOf("пп");

if (index != -1)

{

text = text.Remove(index, 1); // Удаляем один 'п'

}

// Исправление "помешарть" -> "помешать"

index = text.IndexOf("шарть");

if (index != -1)

{

text = text.Remove(index + 4, 1).Insert(index + 4, "а"); // Заменяем 'р' на 'а'

}

// Вывод исправленной строки

Console.WriteLine(text);

Console.ReadKey();

}

}

**6. Расчёт тестовых примеров на ПК (Тестирование)**

**6.1. Параметрический вывод:**

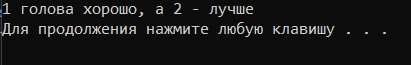


Рисунок 6.1.1 *–* Тестирование параметрического вывода

**6.2. Интерполяция строк:**



Рисунок 6.2.1 *–* Тестирование интерполяции строк

**6.3. Конкатенация строк:**



Рисунок 6.3.1 *–* Тестирование конкатенации строк

**6.4. Строковые методы:**



Рисунок 6.4.1 *–* Тестирование строковых методов

**7. Вывод**

В ходе выполнения работы были реализованы различные способы вывода информации в консоль с использованием языка программирования C#. Параметрический вывод, интерполяция строк, конкатенация и применение строковых методов продемонстрировали возможности гибкой обработки строковых данных. Все задачи были успешно протестированы, и результаты соответствуют ожидаемым. Задачи решены в соответствии с требованиями задания.