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

Ванчаев Д.С.

Вариант №10

19.10.2024

Проверил: к.т.н., доц. Сафронов А.И.

Москва – 2024 г.

**1. Цель работы**

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.

**2.Формулировки задач:**

Формулировка задачи *1*:

Организовать параметрический вывод в консоль сообщения вида, представленного строковой константой. Подобрать подходящие типы данных для перечисленных переменных, основываясь на константах, указанных после математических знаков равенства «=».

"{1} поклон {4}{0} {2} хозяину {5} хозяйкой{0} {3} добрым людям», где

0 – cnstI = ',',

1 – cnstII = 1,

2 – cnstIII = 2,

3 – cnstIV = 3,

4 – cnstV = "Богу",

5 – cnstVI = 'с',

6 – cnstVII = "всем".

Формулировка задачи *2*:

При использовании механизма интерполяции строк (начинается со служебного символа «*$*»), внедрённого в язык *C#*, начиная с версии *Microsoft Visual Studio* от 2015 года ввести конструкцию из переменных, образующих предложение, записанное транслитерацией так, чтобы результирующая строковая конструкция в консоли так же оказалась осмысленным предложением, но другим (иного содержания), записанным кириллицей.

$"{I} {kazhdiy} {poshel} {svoeyu} {dorogoy} {a} {poezd} {ushel} {svoey}"

a = "ни",

dorogoy = '–',

I = "За",

kazhdiy = 2,

poezd = 1,

poshel = "зайцами",

svoey = "поймаешь"

svoeyu = "погонишься",

ushel = "не".

Формулировка задачи *3*:

При выводе каждое слово должно быть символьной константой, каждый служебный символ содержится в символьной переменной и подставляется в итоговую конструкцию через неё, все числовые значения содержатся в целочисленных переменных. Элементы итоговой конструкции должны быть сопряжены посредством конкатенации строк (оператор «+»).

И швец, и жнец, и на дуде игрец.

Формулировка задачи *4*:

Выполните замену одних символов на другие посредством обёртывания результата *Replace()* ещё одним *Replace()*-методом:

"Тр\_пинка, зап\_\_х, жур\_\_вли, д\_\_лёкая, \_сенние, схв\_\_тил, д\_лина"\*

**3. Схема алгоритма**

**3.1. Параметрический вывод**

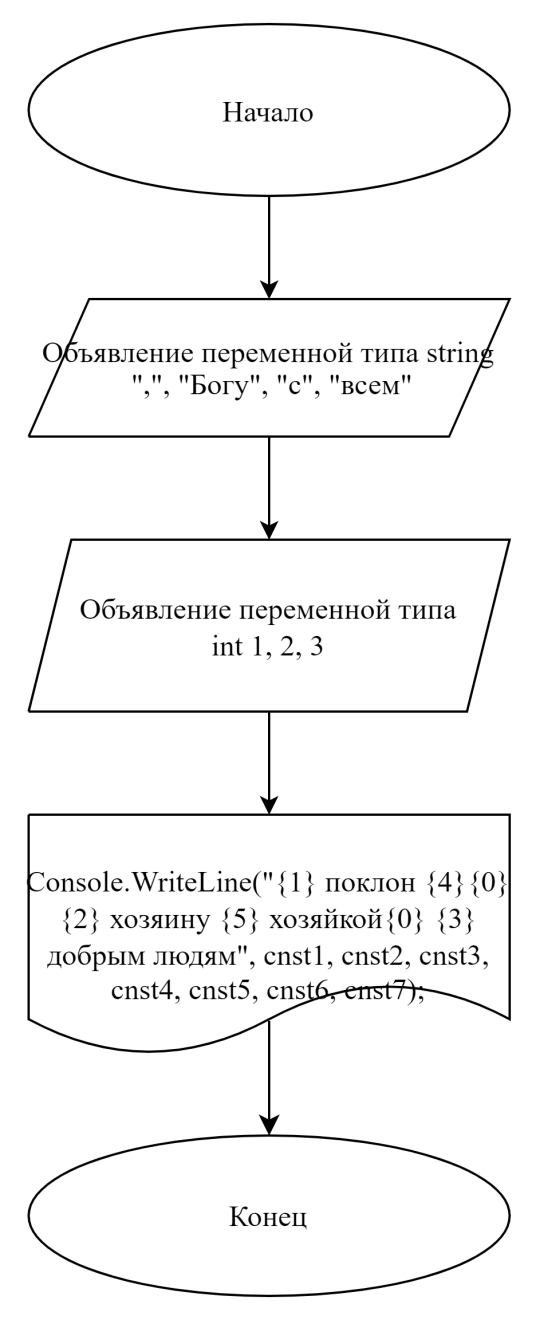


Рисунок 3.1 – Схема алгоритма задачи на константы

**3.2 Интерполяция строк**

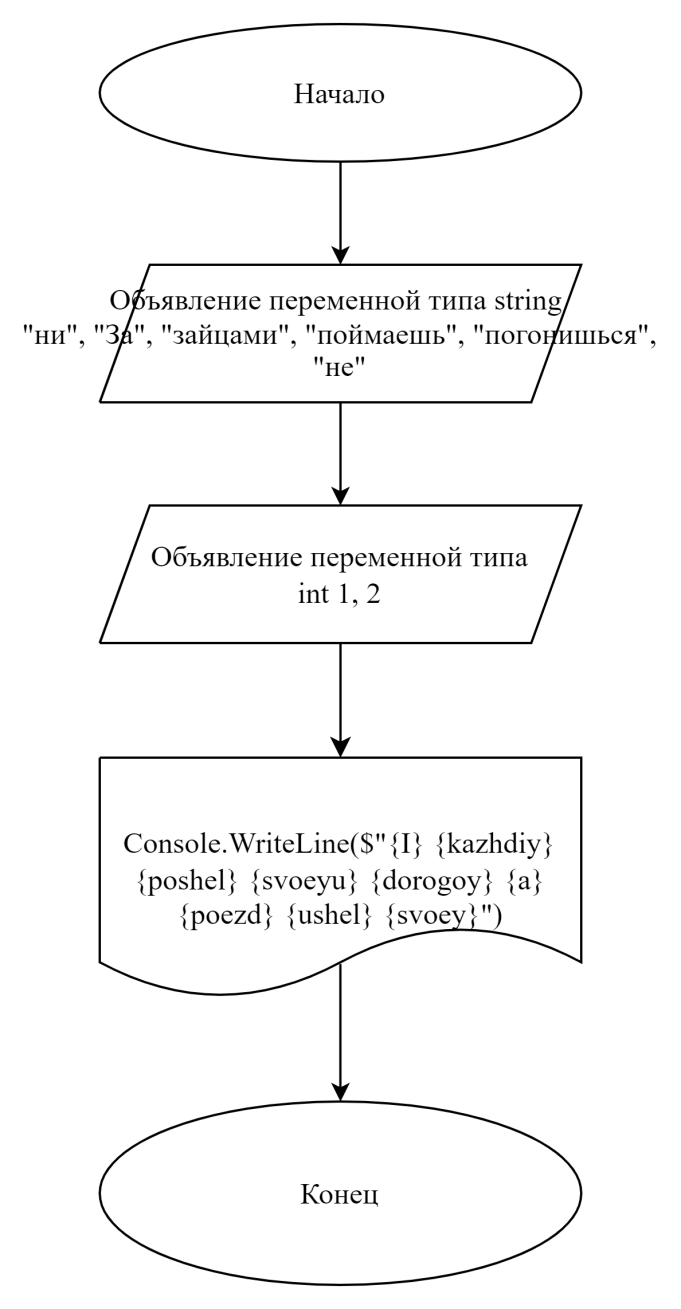


Рисунок 3.2 – Схема задачи на переменные

**3.3. Конкатенация строк**

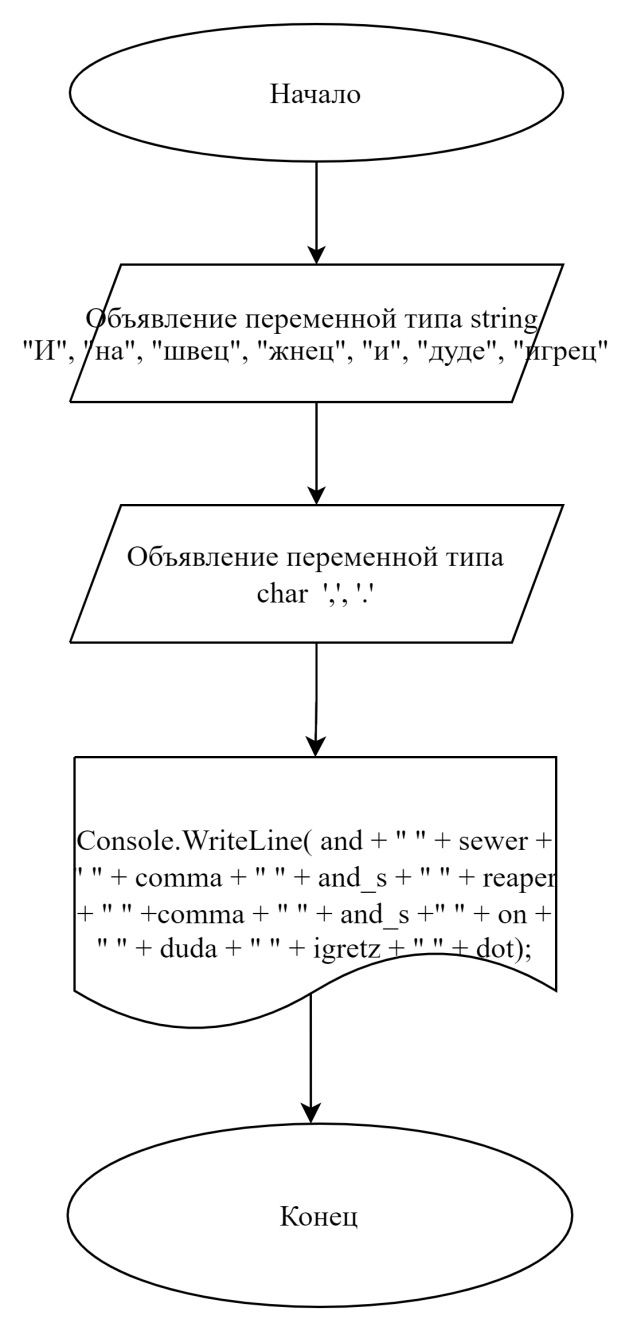


Рисунок 3.3 – Схема алгоритма задачи на конкатенацию

**3.4. Строковые методы**

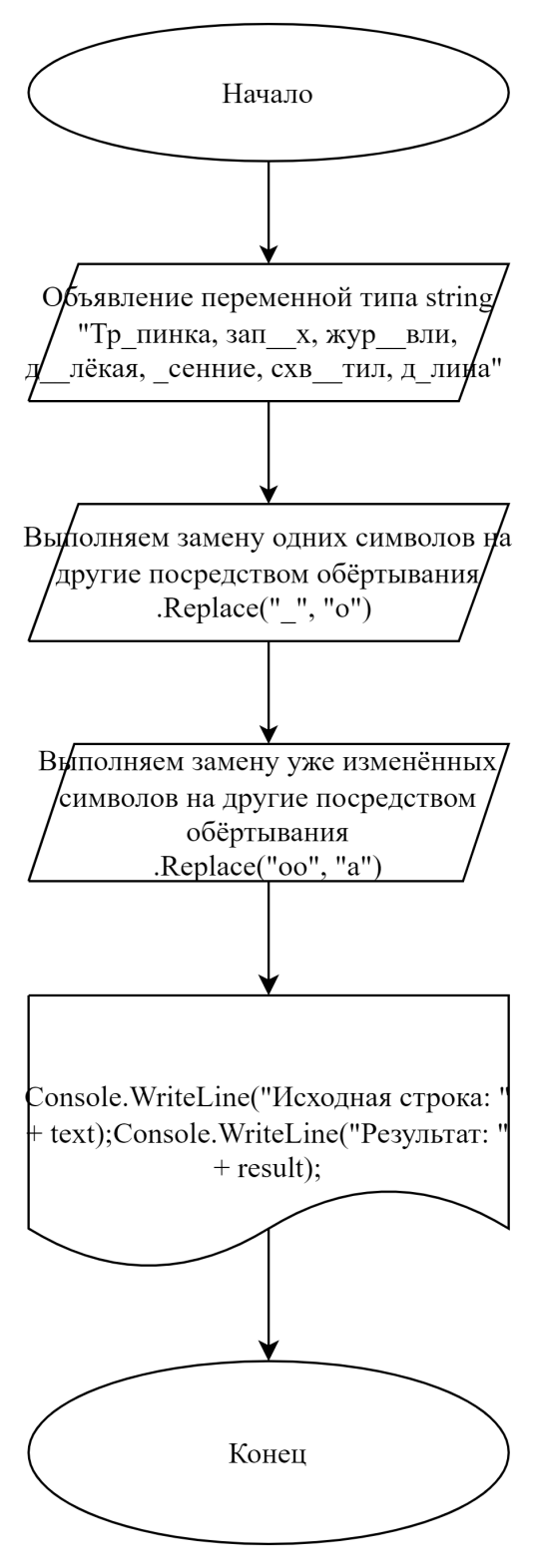


Рисунок 3.4 – Схема алгоритма задачи на конкатенацию

**3.5 Текстовое описание**

**3.5.1 Параметрический вывод**

1. Начало: запускаем программу.
2. Объявление переменных: объявляем переменные, которые будут использоваться для составления строки. Задаем значения:

string cnst1 = ",";

int cnst2 = 1;

int cnst3 = 2;

int cnst4 = 3;

string cnst5 = "Богу";

string cnst6 = "с";

string cnst7 = "всем";

1. Вывод строки в консоль: Используя значения переменных, выводим на экран составленную строку, где каждый параметр подставляется на свое место в выводе.
2. Конец: завершаем выполнение программы.

**3.5.2** **Интерполяция строк**

1. Начало: запускаем программу.

2. Объявление переменных: объявляем переменные для частей строки:

string a = "ни";

char dorogoy = '-';

string I = "За";

int kazhdiy = 2;

int poezd = 1;

string poshel = "зайцами";

string svoey = "поймаешь";

string svoeyu = "погонишься";

string ushel = "не";

1. Вывод строки в консоль: выводим на экран строку, соединяя все переменные с помощью интерполяции, чтобы получить полный текст.
2. Конец: завершаем выполнение программы.

**3.5.3** **Конкатенация строк**

1. Начало: запускаем программу.

2. Объявление переменных: объявляем переменные для частей строки:

string and = "И";

string on = "на";

string sewer = "швец";

string reaper = "жнец";

char comma = ',';

string and\_s = "и";

string duda = "дуде";

string igretz = "игрец";

char dot = '.';

1. Конкатенация и вывод строки: объединяем строки and, on, sewer, …, dot. Выводим результат на экран.
2. Конец: завершаем выполнение программы.

**3.5.4 Строковые методы**

1. Начало: запускаем программу.

2. Объявляем переменные: объявляем переменную ждя строки

String text = "Тр\_пинка, зап\_\_х, жур\_\_вли, д\_\_лёкая, \_сенние, схв\_\_тил, д\_лина"

String result = text

1. Выполняем замену переменных посредством обёртывания:

.Replace("\_", "о")

.Replace("оо", "а");

1. Вывод строки в консоль: выводим на экран итоговую строку с новыми переменными.
2. Конец: завершаем выполнение программы.

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

Тестовая строка:"{1} поклон {4}{0} {2} хозяину {5} хозяйкой{0} {3} добрым людям»

Подставляемые значения:

cnst1 = ",".

cnst2 = 1.

cnst3 = 2.

cnst4 = 3.

cnst5 = "Богу".

cnst6 = "с".

cnst7 = "всем".

Ожидаемый результат: 1 поклон богу, 2 хозяину с хозяйкой, 3 всем добрым людям

* 1. **Интерполяция строк:**

Ввод:

a = "ни".

dorogoy = '-'.

I = "За".

kazhdiy = 2.

poezd = 1.

poshel = "зайцами".

svoey = "поймаешь".

svoeyu = "погонишься".

ushel = "не".

Вывод: За 2 зайцами погонишься - ни 1 не поймаешь

* 1. **Конкатенация строк:**

Исходные данные:

Строка: “И швец, и жнец, и на дуде игрец.”

Ожидаемый результат: И швец, и жнец, и на дуде игрец.

* 1. **Строковые методы:**

Исходная строка: "Тр\_пинка, зап\_\_х, жур\_\_вли, д\_\_лёкая, \_сенние, схв\_\_тил, д\_лина"

Ожидаемый результат: Тропинка, запах, журавли, далёкая, осенние,

схватил, долина

1. **Листинг (код программы)**

**5.1 Параметрический вывод:**

using System;

namespace task\_2\_1

{

class Program

{

public static void Main(string[] args)

{

string cnst1 = ",";

int cnst2 = 1;

int cnst3 = 2;

int cnst4 = 3;

string cnst5 = "Богу";

string cnst6 = "с";

string cnst7 = "всем";

Console.WriteLine("Выводимое константное сообщение...");

Console.WriteLine("{1} поклон {4}{0} {2} хозяину {5} хозяйкой{0} {3} добрым людям", cnst1, cnst2, cnst3, cnst4, cnst5, cnst6, cnst7);

Console.WriteLine("Нажмите любую клавишу для завершения");

Console.ReadKey(true);

}

}

}

**5.2 Интерполяция строк:**

using System;

namespace task\_2\_2

{

class Program

{

static void Main(string[] args)

{

string a = "ни";

char dorogoy = '-';

string I = "За";

int kazhdiy = 2;

int poezd = 1;

string poshel = "зайцами";

string svoey = "поймаешь";

string svoeyu = "погонишься";

string ushel = "не";

Console.WriteLine("Выводимое константное сообщение:");

Console.WriteLine($"{I} {kazhdiy} {poshel} {svoeyu} {dorogoy} {a} {poezd} {ushel} {svoey}");

Console.WriteLine("Нажмите любую клавишу для завершения");

}

}

}

**5.3 Конкатенация строк:**

using System;

namespace task\_2\_3

{

class Program

{

static void Main(string[] args)

{

string and = "И";

string on = "на";

string sewer = "швец";

string reaper = "жнец";

char comma = ',';

string and\_s = "и";

string duda = "дуде";

string igretz = "игрец";

char dot = '.';

Console.WriteLine("Выводимое константное сообщение...");

Console.WriteLine( and + " " + sewer + " " + comma + " " + and\_s + " " + reaper + " " +comma + " " + and\_s +" " + on + " " + duda + " " + igretz + " " + dot);

Console.WriteLine("Нажмите любую клавишу для завершения.");

Console.ReadKey(true);

}

}

}

**5.4** **Строковые методы**

using System;

namespace task\_2\_4

{

internal class Program

{

static void Main(string[] args)

{

string text = "Тр\_пинка, зап\_\_х, жур\_\_вли, д\_\_лёкая, \_сенние, схв\_\_тил, д\_лина";

string result = text

.Replace("\_", "о")

.Replace("оо", "а");

Console.WriteLine("Исходная строка: " + text);

Console.WriteLine("Результат: " + result);

Console.ReadKey(true);

}

}

}

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

**6.1** **Параметрический вывод:**

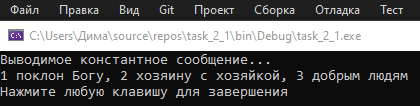


Рисунок 6.1 *–* Тестирование вывода информации параметрическим способом

**6.2 Интерполяция строк:**

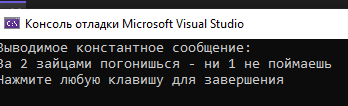


Рисунок 6.2 *–* Тестирование вывода информации с помощью интерполяции строк

**6.3 Конкатенация строк:**

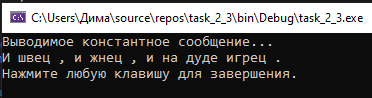


Рисунок 6.3 *–* Тестирование вывода информации с помощью конкатенации строк

**6.4 Строковые методы:**

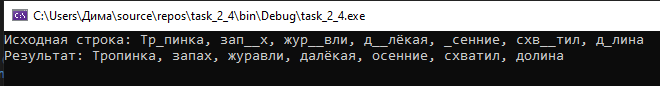


Рисунок 6.4 *–* Тестирование вывода информации строковым методом

**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.