МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ**

**ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**

**(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

**Лабораторная работа №9**

по дисциплине: Основы программирования

тема: «Использование множественного типа»

Выполнил: ст. группы ПВ-201

Машуров Дмитрий Русланович

Проверил:

Притчин Иван Сергеевич

Белгород 2020 г.

**Лабораторная работа №9**

**«Использование множественного типа»**

**Цель работы:** получение навыков работы с множествами

**Задания для подготовки к работе:**

1. Изучите множественный тип данных языка Паскаль.
2. Разбейте задачу соответствующего варианта на подзадачи, таким образом, чтобы решение каждой подзадачи описывалось подпрограммой, а основная программа состояла бы в основном из вызовов подпрограмм.
3. Опишите блок-схему алгоритма решения задачи в укрупненных блоках.
4. Для каждой подзадачи опишите используемые структуры данных, спецификацию и блок-схему алгоритма
5. Опишите блок-схему алгоритма решения задачи с использованием блоков «предопределенный процесс».
6. Закодируйте алгоритм, не используя подпрограммы стандартной библиотеки для обработки строк. Если в библиотеке есть подпрограмма для решения выделенной подзадачи, то следует описать ее самостоятельно, сохранив название, назначение и список параметров стандартной.
7. Подберите наборы тестовых данных с обоснованием их выбора.

**Задания к работе:**

1. Наберите программу, отладьте ее и протестируйте.
2. Выполните анализ ошибок, выявленных при отладке программы.

**Задание варианта №17**

Даны две строки и . Для каждого слова строки вывести слово строки , которое содержит множество гласных букв слова

**Выполнение работы:**

1. **Выделение подзадач**

Выделим следующие подзадачи:

1. Ввод строк s1 и s2
2. Выделение слова из строки s1
3. Сравнение выделенного слова w из строки s1 со словами в строке s2
4. Вывод результата для слова w
5. **Описание структур данных**

ZN – константа, множество знаков, описывающие знаки, которые не нужно учитывать

VOWEL – константа, множество гласные букв, которые нужно считать

**const**

ZN: **set of** char = [' ','.',',',':','?','!'];

VOWEL: **set of** char ['а','о','у','э','ы','я','е','ю','и','А','О','У','Э','Ы','Я','Е','Ю','И'];

1. **Блок-схема алгоритма в укрупнённых блоках**

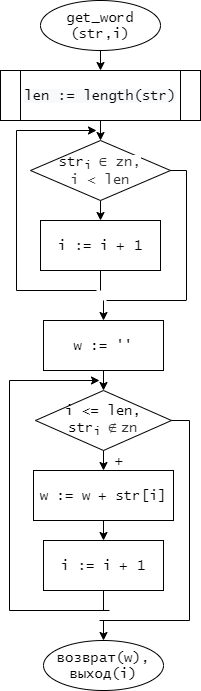
****

1. **Описание подпрограмм**

Спецификация функции get\_word

1. Заголовок: function get\_word(str: string; var i: integer) : string
2. Назначение: возвращает слово из строки str начинающееся с индекса i. Параметру i присваивает номер символа после считанного слова
3. Входные параметры: str, i
4. Выходные параметры: w

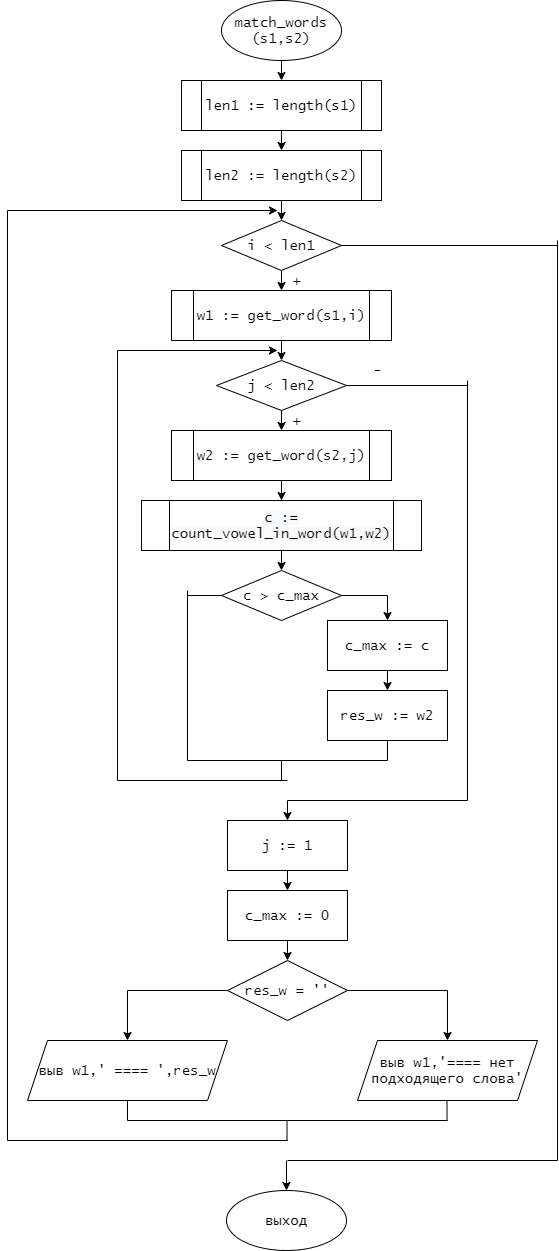
Блок-схема:



Спецификация функции match\_words

1. Заголовок procedure match\_words(s1: string; s2: string)
2. Назначение: Для каждого слова строки s1 вывод слова строки s2, которое содержит множество гласных букв слова строки s1
3. Входные параметры: s1, s2
4. Выходные параметры: нет

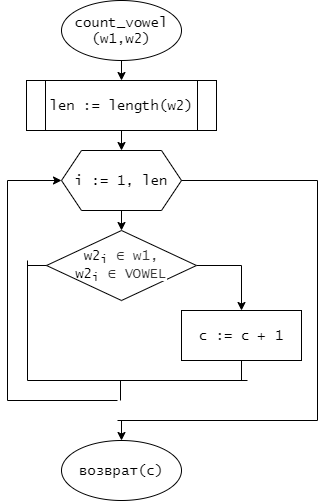
Блок-схема:



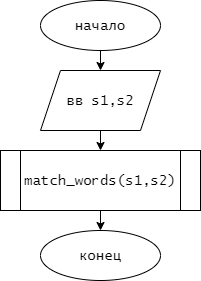
Спецификация функции count\_vowel

1. Заголовок: function count\_vowel(w1: string; w2: string) : integer
2. Назначение: возвращает количество гласных букв слова w2, которые содержатся в слове w1
3. Входные параметры: w1, w2
4. Выходные параметры: c

Блок-схема:



1. **Блок-схема с блоками «предопределённый процесс»**

****

1. **Тестовые данные**

|  |  |  |
| --- | --- | --- |
| № п/п | Предложения | Результат |
| 1 | Значимость этих проблем настолько очевидна | Значимость = организации  Этих = организации  Проблема = особенности  Настолько = задача  Очевидна = организации |
| Задача организации, в особенности же |
| 2 | Не следует, однако, забывать о... | Не = намеченного  Следует = намеченного  Однако = намеченного  Забывать = реализация  О = Дорогие |
| Дорогие друзья, реализация намеченного плана... |
| 3 | Соображения высшего порядка, а... | Соображения = очевидна  Высшего = проблема  Порядка настолько |
| Значимость этих проблем настолько очевидна |

1. **Текст программы**

**const**

ZN: **set of** char = [' ','.',',',':','?','!'];

VOWEL: **set of** char = ['а','о','у','э','ы','я','е','ю','и','А','О','У','Э','Ы','Я','Е','Ю','И'];

**function** get\_word(str: string; **var** i: integer) : string;

**var** w: string;

len: integer;

**begin**

len := length(str);

**while** (str[i] **in** zn) **and** (i < len) **do**

i := i + 1;

w := '';

**while** (i <= len) **and not**(str[i] **in** zn) **do**

**begin**

w := w + str[i];

i := i + 1;

**end**;

get\_word := w;

**end**;

**function** count\_vowel(w1: string; w2: string) : integer;

**var** i, len:integer;

c: integer;

**begin**

len := length(w2);

**for** i := 1 **to** len **do**

**if** (w2[i] **in** w1) **and** (w2[i] **in** VOWEL) **then**

c := c + 1;

count\_vowel := c;

**end**;

**procedure** match\_words(s1: string; s2: string);

**var**

i, j, len1, len2, c, c\_max: integer;

w1, w2, res\_w: string;

**begin**

i := 1;

j := 1;

len1 := length(s1);

len2 := length(s2);

**while** (i < len1) **do**

**begin**

w1 := get\_word(s1,i);

**while** (j < len2) **do**

**begin**

w2 := get\_word(s2,j);

c := count\_vowel(w1,w2);

**if** (c > c\_max) **then**

**begin**

c\_max := c;

res\_w := w2;

**end**;

**end**;

j := 1;

c\_max := 0;

**if** (res\_w = '') **then**

writeln(w1,' ==== нет подходящего слова')

**else**

writeln(w1,' ==== ',res\_w);

**end**;

**end**;

**var**

s1, s2: string;

**begin**

writeln('Введите первую строку');

readln(s1);

writeln('Введите вторую строку');

readln(s2);

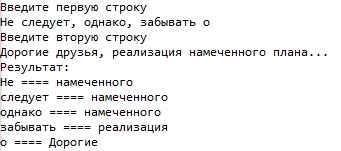
writeln('Результат:');

match\_words(s1,s2);

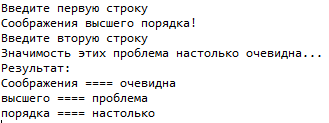
**end**.

1. **Результаты работы программы**

*Пример 1:*

******

*Пример 2:*

******

1. **Анализ допущенных ошибок**

* Забыл обнулить c\_max после прохода по строке
* Забыл вернуть значение j к началу строки