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 №6**

по дисциплине: Основы программирования

тема: «Обработка символьных строк»

Выполнил: ст. группы ПВ-201

Машуров Дмитрий Русланович

Проверил:

Притчин Иван Сергеевич

Белгород 2020 г.

**Лабораторная работа №6**

**«Обработка символьных строк»**

**Цель работы:** получение навыков работы со строками. Закрепление навыков работы с подпрограммами.

**Задания для подготовки к работе:**

1. Изучите строковый тип, организацию работы со строками в Паскале; стандартные строковые функции.
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**

Определить, есть ли в данной строке одинаковые слова

**Выполнение работы:**

1. **Выделение подзадач**

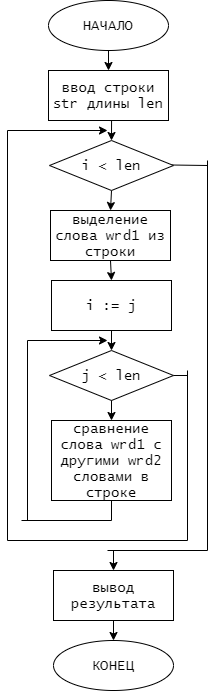
Выделим следующие подзадачи:

1. Ввод строки
2. Выделение слова из строки
3. Сравнение слова с другими словами в строке
4. Вывод результата
5. **Описание структур данных**

zn – константа, множество знаков, которые не нужно учитывать в разбиении на строки

**const** zn = [' ','.',',',':','?','!'];

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

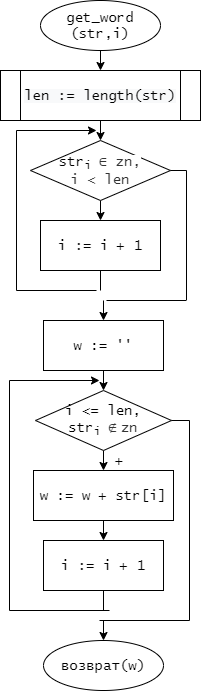
****

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
3. Входные параметры: str, i
4. Выходные параметры: w

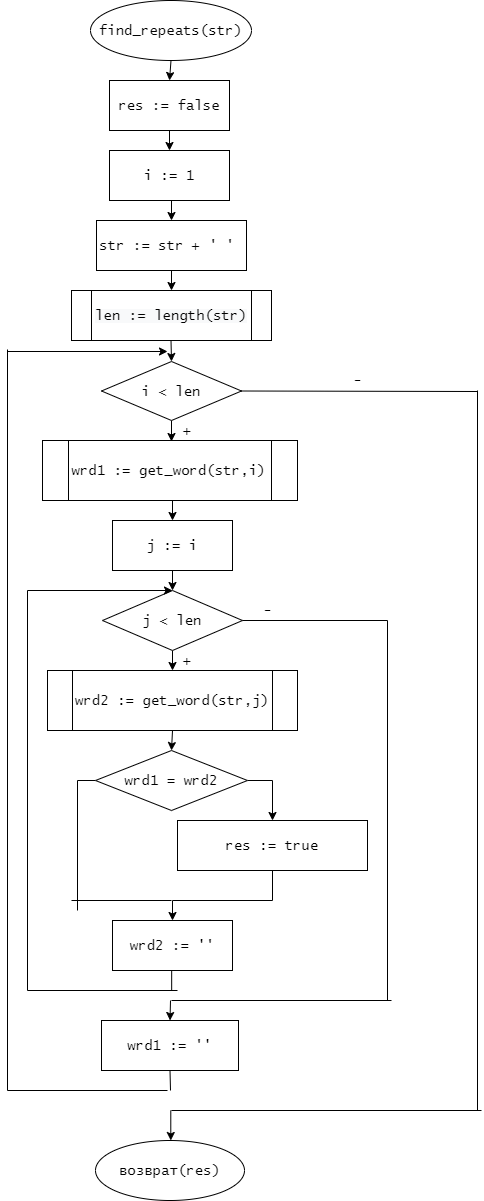
Блок-схема:



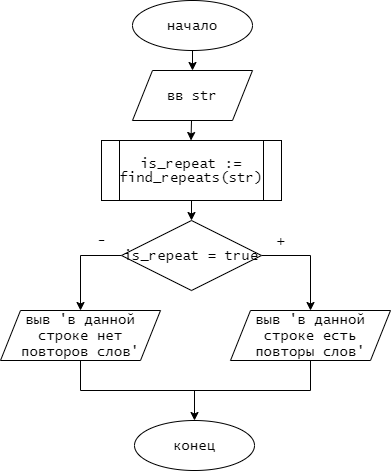
Спецификация функции find\_repeats

1. Заголовок function find\_repeats(str: string)
2. Назначение: возвращает “истину”, если в строке str есть повторяющиеся слова, иначе – “ложь”
3. Входные параметры: str
4. Выходные параметры: res

Блок-схема:



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



1. **Тестовые данные**

|  |  |  |
| --- | --- | --- |
| № | Строка | Результат |
| 1 | Съешь ещё этих мягких французских булок, да выпей же чаю | Повторов нет |
| 2 | Слово……………………ещё слово……………………….сло во, которое повторяет первое слово…. | Повторы есть |
| 3 | Когда один, тогда не два, а один… | Повторы есть |

1. **Текст программы**

**const**

zn = [' ','.',',',':','?','!'];

**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** find\_repeats(str: string) : boolean;

**var** i,j: integer;

wrd1, wrd2: string;

len: integer;

res: boolean;

**begin**

res := false;

i := 1;

str := str + ' ';

len := length(str);

**while** i < len **do**

**begin**

wrd1 := get\_word(str,i);

j := i;

**while** (j < len) **do**

**begin**

wrd2 := get\_word(str,j);

**if** (wrd1 = wrd2) **then**

res := true;

wrd2 := '';

**end**;

wrd1 := '';

**end**;

find\_repeats := res;

**end**;

**var**

str: string;

flag: boolean;

is\_repeat: boolean;

**begin**

writeln('Введите строку');

read(str);

is\_repeat := find\_repeats(str);

**if** (is\_repeat = true) **then**

writeln('В данной строке есть повторы слов')

**else** writeln('В данной строке нет повторов слов')

1. **Результаты работы программы**

*Пример №1:*



*Пример №2:*



*Пример №3:*



1. **Анализ допущенных ошибок**

* Упущена ; в конце
* В конечном условии результат функции find\_repeats сравнивался со значением “ложь”