**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №2**

**по дисциплине «Программирование»**

Тема: *Управляющие конструкции языка Си*

|  |  |  |
| --- | --- | --- |
| Студент гр. 6303 |  | Горбунова А.П. |
| Преподаватель |  | Берленко Т.А. |

Санкт-Петербург

2016

Цель: написать программу, которая форматирует некоторый текст и выводит результат на консоль.

Ход работы:

1. В главной функции «main» инициализируем переменные, вызываем функцию «read» и выводим количество предложений до работы программы («aft + que») и количество предложений после работы программы («aft»).

int main()

{

int result = 0; //вспомогательная переменная

int aft = 0; //количество предложений после работы программы

int que = 0; //количество вопросительных предложений

while (result == 0)

read(&result, &aft, &que);

printf("Количество предложений до %d и количество предложений после %d", (aft + que), aft);

return 0;

}

1. В функции «read» инициализируем массив символов и выделяем под него память с помощью malloc, предварительно подключив в препроцессоре библиотеку <stdlib.h>

char\* massiv = (char\*)malloc(100 \* sizeof(char));

1. Воспользуемся циклом while. Он считывает символы до тех пор, пока не встретит точку, точку с запятой или вопросительный знак. В массив записываются символы, не являющиеся табуляцией, символом переноса строки или пробелом.

Если встречается символ восклицательного знака, вспомогательная переменная становится равной единице, в последующем мы этим воспользуемся.

while (((ch = getchar()) != '.') && (ch != '?') && (ch != ';'))

{

if (ch == '!')

{

\*result = 1;

break;

}

if (((ch != '\t') && (ch != '\n') && (ch != ' ')) || (i != 0))

massiv[i++] = ch;

}

1. Если вспомогательная переменная не изменилась, используем цикл if, чтобы добавить знаки препинания в конец предложения.

Если встречается символ, равный точке или точке с запятой, то записываем его в массив и добавляем символ конца строки, затем выводим массив символов, добавляя в переменную aft единицу, тем самым подсчитывая количество невопросительных предложений, то есть предложений после работы программы. Если такого символа не встречается, то добавляем единицу в переменную que, то есть подсчитываем вопросительные предложения.

if (\*result == 0)

if ((ch == '.') || (ch == ';'))

{

massiv[i++] = ch;

massiv[i] = '\0';

printf("%s\n", massiv);

++\*aft;

}

else

++\*que;

}

**Вывод:** в ходе лабораторной работы получены навыки работы с указателями и управляющими конструкциями языка Си, которые успешно применены на практике при создании программы.