

Отчет по лабораторной работе № 11 по курсу “Фундаментальная информатика”

Студент группы М80-109Б-22 Воропаев Иван Константинович, № по списку
2

Контакты e-mail, vk.com/okteamail, tg: @Nyamerka

Работа выполнена: «2» декабря 2022г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

Отчет сдан « » _____ 20__ г., итоговая оценка

Подпись преподавателя

1. **Тема:** Обработка последовательности литер входного текстового файла. Простейшие приёмы лексического анализа. Диаграммы состояний и переходов.
2. **Цель работы:** Составить программу на языке Си, выполняющую анализ и обработку вводимого текста в соответствии с выданным преподавателем вариантом задания.
3. **Задание:** Выделить десятичные целые числа без знака и напечатать суммы их чётных чисел.
4. **Оборудование:** MacBook Pro “13” 2017г, 4 порта Thunderbolt
5. **Программное обеспечение:** MacOS Ventura 13.0
6. **Идея, метод, алгоритм решения задачи:** Создать машину состояний, выявляющую из входных данных слов, являющихся целыми десятичными числами.
7. **Сценарий выполнения работы**
 - Придумать машину состояний, выявляющую из входного слова десятичное целое число.
 - Осуществить печать нужных данных.
 - Провести UNIT-тестирование, исходный код дестов приложен в пункте 8.
 - Защитить лабу.

8. Распечатка протокола:

```
#include <stdlib.h>
#include <stdio.h>
#include <assert.h>
#include <stdbool.h>

typedef enum {
    NewWord,
    Rubbish,
    Number
} State;

bool is_NewWord (char symbol) {
    if (symbol == '\n' || symbol == '\t' || symbol == ';' || symbol == ' ') {
        return true;
    }
    return false;
}

bool is_Rubbish (char symbol) {
    if ((symbol < '0' || symbol > '9') && is_NewWord(symbol) == false) {
        return true;
    }
    return false;
}

bool is_PlusMinus (char symbol) {
    if (symbol == '+' || symbol == '-') {
        return true;
    }
    return false;
}

bool is_Number (char symbol) {
    if (symbol >= '0' && symbol <= '9') {
        return true;
    }
    return false;
}

void printArray (int n, char array [n]) {
    for (int i = 0; i < n; i++) {
        printf("%c", array [i]);
    }
    printf(" ");
}

int main () {
    assert(is_Rubbish('a') == true);
    assert(is_Rubbish('@') == true);
    assert(is_Rubbish('0') == false);
}
```

```

assert(is_NewWord('\n') == true);
assert(is_NewWord('\t') == true);
assert(is_NewWord('i') == false);
assert(is_PlusMinus('+') == true);
assert(is_PlusMinus('a') == false);
assert(is_Number('0') == true);
assert(is_Number('a') == false);
State state = NewWord;
char symbol, *number;
number = (char *) malloc(1);
int i = 0, summ = 0, firstZero = 0;
bool flag = true;
while ((symbol = getchar()) != EOF) {
    switch (state) {
        case NewWord:
            if (is_PlusMinus(symbol)) {
                state = Number;
                break;
            }
            if (is_Rubbish(symbol)) {
                state = Rubbish;
                break;
            }
            if (is_Number(symbol)) {
                state = Number;
                firstZero = symbol - '0';
                if (firstZero == 0) {
                    break;
                }
            }
            else {
                flag = false;
            }
            number[i] = symbol;
            i++;
            number = (char *) realloc(number, i + 1);
            break;
        case Rubbish:
            if (is_NewWord(symbol)) {
                state = NewWord;
            }
            break;
        case Number:
            if (is_Number(symbol)) {
                if (flag) {
                    firstZero = symbol - '0';
                    if (firstZero == 0) {
                        flag = true;
                        break;
                    }
                }
                else {
                    flag = false;
                }
            }
    }
}

```

```
        }
        number[i] = symbol;
        if(i % 2 != 0) {
            summ += symbol - '0';
        }
        i++;
        number = (char *) realloc(number, i + 1);
        break;
    }
    if (is_NewWord(symbol)) {
        state = NewWord;
        printArray(i, number);
        printf("%d\n", summ);
    }
    else {
        state = Rubbish;
    }
    i = 0;
    summ = 0;
    flag = true;
    firstZero = 0;
    break;
}
return 0;
}
```

9. **Дневник отладки:**

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание
0	Дом	2.12.2022	18:33	Доделал лабу. Намана.		Вроде всё учёл.

10. **Замечания автора:** Не сохраняет входные данные.

11. **Выводы:** В ходе работы я научился осуществлять обработку последовательности литер входного файла с помощью машины состояний, что позволит мне в будущем осуществлять алгоритмы более сложных обработок.

Подпись студента _____