

# Отчет о лабораторной работе №3

## Описание func.cpp

1. `void convertToArray(bool fileOrConsole, char convertedText[20][20][40])`

### Входные данные:

(наличие флага “—fromFile” при запуске программы, 3-х мерный массив, куда будет записан введенный текст.)

### Примечания:

Используется 3х мерный массив для хранения введенного текста. То есть массив – состоящий из предложений, каждое из которых состоит из слов, каждое слово из которых состоит из символов.

Функция разделена на 2 части, 1ая отвечающая за обработку текста из консоли, 2ая отвечающая за обратотку текста из файла. Алгоритм и там, и там одинаковый и будет описан далее. Различие лишь в способе получения данных, используются разные функции. В 1ом случае используется `getchar()`, во 2ом `text.get()`, где `text` – объект `std::ifstream`. Также различное условие выхода из цикла. В 1ом случае `while(character!='\n');`, во 2ом `while(!text.eof());`.

### Описание алгоритма:

1. Считываем символ
2. Если символ – пробел, то добавляем его в текущее слово, увеличиваем счетчик слов на 1, обнуляем кол-во символов в слове, т.к далее будем записывать уже новое слово. Также проверяем, что последний элемент был точкой, в этом случае просто пропускаем 1 итерацию цикла.
3. Если символ – не точка, то добавляем его в слово, увеличиваем счетик кол-ва символов в слове.
4. Если символ – точка, то запоминаем, что последний элемент – точка. Добавляем `&` в конец слова (в данной программе `&` обозначает окончание предложения). Увеличиваем счечик кол-ва предложений, обнуляем счетчик слов в предложении, символов в слове, т.к переходим к записи нового предложения.
5. Если символ – конец строки и уже встречалась точка, то добавляем в следующее предложение в первое слово на 1ую позицию `&`. Будем считать, что если амперсant встречается именно в этом месте, то здесь обозначено окончание ввода/ файла.
6. Если символ- конец строки, но точки еще не было, то просто переходим к следующей итерации цикла.

```
1 void convertToArray(bool fileOrConsole, char convertedText[20][20][40]){ //file=1;console=0
2     bool lever = false;
3     std::ifstream text;
4     if (fileOrConsole) {
5         text.open("FileWithText.txt");
6         if(text.is_open()){
7             lever=true;
8         }
9     }
10    char character;
11    int sentenceCount=0,sentenceWordCount=0,wordCharactersCount=0;
12    bool lastWasDot = false, dotIsHere = false;
13    if(!lever){
14        do{
15            if(lever) character= text.get();
16            character = getchar();
17            if(character == ' '){
```

```

18     if(lastWasDot){
19         lastWasDot=false;dotIsHere=true;continue; //пропускаем пробел после .
20     }
21     convertedText[sentenceCount][sentenceWordCount][wordCharactersCount] = character;
22     wordCharactersCount = 0;
23     sentenceWordCount++;
24     continue;
25 }
26 if(character!='.'){
27     convertedText[sentenceCount][sentenceWordCount][wordCharactersCount] = character;
28     wordCharactersCount++;
29 }
30 if(character=='.'){
31     lastWasDot=true;
32     dotIsHere=true;
33     convertedText[sentenceCount][sentenceWordCount][wordCharactersCount] = '&';
34     // &=end of sentence
35     wordCharactersCount=0;
36     sentenceWordCount=0;
37     sentenceCount++;
38 }
39 }
40 if((dotIsHere)&&(character=='\n')){
41     convertedText[sentenceCount][0][0]='&';
42 }
43 if(!dotIsHere){
44     character = 'I';
45     continue;
46 }
47 }
48 }while(character!='\n');
49 }
50 if (lever){
51     do{
52         character= text.get();
53         if(character == ' '){
54             if(lastWasDot){
55                 lastWasDot=false;dotIsHere=true;continue; //пропускаем пробел после .
56             }
57             convertedText[sentenceCount][sentenceWordCount][wordCharactersCount] = character;
58             wordCharactersCount = 0;
59             sentenceWordCount++;
60             continue;
61         }
62         if((character!='.')){
63             convertedText[sentenceCount][sentenceWordCount][wordCharactersCount] = character;
64             wordCharactersCount++;
65         }
66         if(character=='.'){
67             lastWasDot=true;
68             dotIsHere=true;
69             convertedText[sentenceCount][sentenceWordCount][wordCharactersCount] = '&';
70             // &=end of sentence
71             wordCharactersCount=0;
72             sentenceWordCount=0;
73             sentenceCount++;
74         }
75     }
76     if((dotIsHere)&&(character=='\n')){
77         convertedText[sentenceCount][0][0]='&';
78     }
79     if(!dotIsHere){

```

```

80         character = '1';
81         continue;
82     }
83
84     }while(!text.eof());
85     convertedText[sentenceCount+1][0][0]='&';
86 }
87 }

```

**Примечание:** Функция void strangePrint(...) не будет описана, т.к использовалась только для отладки.

2. void countLength(int Length[20],char text[20][20][40]){

#### Входные данные:

(массив, который будет хранить длину каждого предложения, полученный текст)

#### Описание алгоритма:

1. Обнуляем массив длин, чтобы избежать дальнейших ошибок.
2. Далее проходимся по предложениям в полученном тексте.
3. Если встречаем символ – конец предложение (описание условия конца предложения описано в 1ой функции func.cpp), то выходим из цикла
4. Проходим по словам в каждом предложении
5. Проходим по символам в каждом слове. Далее проверяем условия конца слова(выходим из цикла), либо наличия символа переноса строки(пропускаем итерацию), либо наличие символа-конца предложения(в этом случае выходим из цикла на 1 уровень выше, с помощью логической переменной)
7. Если все условия удовлетворены, то прибавляем единицу к длине.

```

1 void countLength(int Length[20],char text[20][20][40]){
2     bool brake = false;
3     for(int i=0;i<20;i++){Length[i]=0;}
4     for(int k=0;k<20;k++){
5         if(text[k][0][0]=='&') {break;}
6         for(int j=0;j<20;j++){
7             for(int i=0;i<40;i++){
8                 if(text[k][j][i]=='\0') break;
9                 if(text[k][j][i]=='&') {brake=true;break;}
10                if(text[k][j][i]=='\n'){continue;}
11                Length[k]++;
12            }
13            if (brake) {brake=false;break;}
14        }
15        brake = false;
16    }
17 }

```

3. void printSentence(int SentenceNumber,char text[20][20][40],bool toFile,std::ofstream &answersFile){

#### Входные данные:

( номер предложения, которое необходимо вывести ; полученный текст ; проверка наличия флага "--toFile"; файл, в который будет необходимо произвести запись в случае наличия флага)

#### Описание алгоритма:

1. Проходим по словам в предложении
2. Проходим по символам в каждом слове
3. Проверяем необходимые условия выхода из цикла (описание условий можно найти в описании предыдущей функции)
4. Выводим символ в консоль/ записываем его в файл в зависимости от ситуации.
5. В конце выводим/записываем точку.

```

1 void printSentence(int SentenceNumber,char text[20][20][40],bool toFile,std::ofstream &answersFile){
2     bool brake = false;

```

```

3  for(int i=0;i<20;i++){
4      for(int j=0;j<40;j++){
5          if(text[SentenceNumber][i][j]=='\0') break;
6          if(text[SentenceNumber][i][j]=='&') {brake=true;break;}
7          if(text[SentenceNumber][i][j]=='\n'){continue;}
8
9          if(!toFile){std::cout<<text[SentenceNumber][i][j];}
10         if(toFile){answersFile<<text[SentenceNumber][i][j];}
11     }
12     if (brake) {brake=false;break;}
13 }
14 if(!toFile){std::cout<<'. '<<std::endl;}
15 if(toFile){answersFile<<'. '<<'\n';}
16 }

```

4. `void printReverseSentence(int SentenceNumber, char text[20][20][40]`

**Входные данные:**

(номер предложения с которым ф-я будет работать ; полученный текст)

**Описание алгоритма:**

1. Создаем и очищаем файл, куда будет записано предложение в обратном порядке слов.
2. Далее используем цикла, сильно похожий на циклы, которые были ранее использованы в функциях 2 и 3(отличие в том, что в этом случае мы ищем символ конца предложения)
3. Запоминаем номер слова, где заканчивается предложение.
4. Далее проходим по полученному предложению, начиная с последнего слова, до первого и выводим их поочередно в файл.

```

1  void printReverseSentence(int SentenceNumber, char text[20][20][40]){
2      std::ofstream file("NINE.txt");
3      file.clear();
4      bool brake = false;
5      int storage = 0;
6      for(int i=19;i>=0;i--){
7          for(int j=0;j<40;j++){
8              if(text[SentenceNumber][i][j]=='&'){
9                  storage=i;
10                 brake = true;
11             }
12         }
13         if (brake) {brake=false;break;}
14     }
15
16     for(int i=storage;i>=0;i--){
17         for(int j=0;j<40;j++){
18             if(text[SentenceNumber][i][j]=='\0') break;
19             if(text[SentenceNumber][i][j]=='&') {file<<' ';continue;}
20             if(text[SentenceNumber][i][j]=='\n'){continue;}
21             file<<text[SentenceNumber][i][j];
22         }
23     }
24 }

```

5. `void printAnswer(int Length[20], char text[20][20][40], bool toFile)`

**Входные данные:**

(массив длин каждого предложения; полученный текст; наличие флага “toFile”)

**Описание алгоритма:**

1. Проходим по массив длин несколько раз, запоминаем расположение максимального и минимального числа, в этом массиве, соответственно таким образом находим расположение максимального и минимального по длине предложения в полученном тексте.
2. Выводим предложения в порядке неубывания в консоль/файл
3. Выводим в отдельный файл наибольшее предложение в обратном порядке слов.

```

1 void printAnswer(int Length[20],char text[20][20][40],bool toFile){
2     int minStorage = 0;
3     int maxStorage = 0;
4     int minimum = 100;
5     int maximum = 0;
6     bool check = false;
7     std::ofstream answersFile("Answers.txt");
8     for(int i=0;i<20;i++){
9         for(int j=0;j<20;j++){
10             if(Length[j]==0) continue;
11             if(Length[j]>maximum){maximum=Length[j];maxStorage=j;}
12             if(Length[j]<= minimum){
13                 minStorage=j;
14                 minimum=Length[j];
15                 check = true;
16             }
17         }
18         if(check){
19             minimum=100;
20             Length[minStorage]=0;
21             check = false;
22             printSentence(minStorage,text,toFile,answersFile);
23         }
24     }
25     printReverseSentence(maxStorage,text);
26 }

```

## Описание main.cpp

1. Проверяем наличие флагов
2. Получаем текст из файла/консоли с помощью convertToArray(...)
3. Составляем массив длин с помощью countLength(...)
4. Выводим ответ с помощью printAnswer(...)

## Тест 1.

Текст:

Alice's Adventures in Wonderland is a children's novel. The title is usually shortened to Alice in Wonderland. Charles Lutwidge Dodgson wrote the book. He wrote it using the pen name Lewis Carroll. John Tenniel drew the 42 pictures in the book. The book was published by Macmillan and Co in London. It was released on 26 November 1865. It has been adapted to numerous movies. In 2010, it has been adapted to Hollywood movie.

Ответ:


1. Консоль

```

It was released on 26 November 1865.
It has been adapted to numerous movies.
Charles Lutwidge Dodgson wrote the book.
He wrote it using the pen name Lewis Carroll.
John Tenniel drew the 42 pictures in the book.
In 2010, it has been adapted to ПП0Hollywood movie.
The book was published by Macmillan and Co in London.
The title is usually shortened to Alice in Wonderland.
Alice's Adventures in Wonderland is a children's novel.

```

## 2. NINE.txt


 NINE.txt - Notepad

File Edit Format View Help

novel children's a is Wonderland in Adventures Alice's

## Тест2:

### Текст:


 FileWithText.txt - Notepad

File Edit Format View Help

The Smiths live in a house. They have a living room. They watch TV in the living room. The father cooks food in the kitchen. They eat in the dining room. The house has two bedrooms. They sleep in the bedrooms. They keep their clothes in the closet. There is one bathroom. They brush their teeth in the bathroom.

### Ответ:


#### 1. Answers.txt

 Answers.txt - Notepad

File Edit Format View Help

There is one bathroom.  
They have a living room.  
They sleep in the bedrooms.  
The house has two bedrooms.  
The Smiths live in a house.  
They eat in the dining room.  
They watch TV in the living room.  
The father cooks food in the kitchen.  
They keep their clothes in the closet.  
They brush their teeth in the bathroom.

#### 2. NINE.txt

 NINE.txt - Notepad

File Edit Format View Help

bathroom the in teeth their brush They

## Вывод:

Программа работает корректно, выдает правильные ответы.