УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе №3.1

по предмету «Основы алгоритмизации и программирования»

Вариант 15

Выполнила:

Городко К. Е.

Гр. 351005

Проверила:

Данилова Г. В.

Минск 2023

**ЗАДАНИЕ:**

В заданном тексте в каждом четном слове заменить все строчные

буквенные символы на прописные, а каждое нечетное слово заключить

в круглые скобки.

**КОД ПРОГРАММЫ DELPHI:**

Program LAB3\_1;

Uses

System.SysUtils;

Const

NUMBERS = ['0' .. '9'];

LETTERS = ['A' .. 'Z', 'a' .. 'z', 'А' .. 'Я', 'а' .. 'я', 'Ё', 'ё'];

OPENBRACE = '(';

CLOSEBRACE = ')';

Procedure WriteCondition();

Begin

Writeln('Данная программа в каждом четном слове текста заменяет все буквы на

прописные,', #10, 'а каждое нечетное слово заключает в круглые

скобки.');

End;

Function CheckChoiceInput(OutputMessage: String): Integer;

Var

Num: Integer;

IsCorrect: Boolean;

Begin

Repeat

IsCorrect := True;

Writeln(OutputMessage);

Try

Readln(Num);

Except

IsCorrect := False;

Writeln('Введенные данные не соответствуют условию. Повторите попытку.');

End;

If IsCorrect And ((Num <> 0) And (Num <> 1)) Then

Begin

IsCorrect := False;

Writeln('Введенные данные не соответствуют условию. Повторите попытку.');

End;

Until (IsCorrect);

Writeln;

CheckChoiceInput := Num;

End;

Function IsTextCorrect(Text: AnsiString): Boolean;

Var

I: Integer;

IsCorrect: Boolean;

Begin

IsCorrect := False;

If Text = '' Then

Writeln('В тексте нет символов. Повторите попытку.')

Else

Begin

I := 1;

While Not IsCorrect And (I < (Length(Text) + 1)) Do

Begin

If Text[I] In LETTERS Then

IsCorrect := True;

Inc(I);

End;

If Not IsCorrect Then

Writeln('В тексте нет букв. Повторите попытку.');

End;

IsTextCorrect := IsCorrect;

End;

Function InputTextFromConsole(): AnsiString;

Var

Text: AnsiString;

Begin

Repeat

Writeln('Введите текст.');

Readln(Text);

Until IsTextCorrect(Text);

Writeln;

InputTextFromConsole := Text;

End;

Function IsFilePathCorrect(Path: String): Boolean;

Var

IsCorrect: Boolean;

Begin

IsCorrect := True;

If Not FileExists(Path) Then

Begin

Writeln('Введенного файла не существует. Повторите попытку.');

IsCorrect := False;

End

Else If ExtractFileExt(Path) <> '.txt' Then

Begin

Writeln('Введенный Вами файл не является текстовым. Повторите попытку.');

IsCorrect := False;

End;

IsFilePathCorrect := IsCorrect;

End;

Function CheckFileInputPath(): String;

Var

Path: String;

Begin

Repeat

Writeln('Введите путь к файлу, содержащему текст.');

Readln(Path);

Until IsFilePathCorrect(Path);

CheckFileInputPath := Path;

End;

Function GetTextFromFile(Path: String): AnsiString;

Var

I: Integer;

Text, Line: AnsiString;

IsCorrect: Boolean;

FIn: TextFile;

Begin

Repeat

IsCorrect := True;

Try

Try

AssignFile(FIn, Path);

Reset(FIn);

While Not Eof(FIn) Do

Begin

Readln(FIn, Line);

Text := Text + Line + #10;

End;

IsCorrect := IsTextCorrect(Text);

Finally

CloseFile(FIn);

End;

Except

Writeln('Произошла ошибка. Повторите попытку.');

IsCorrect := False;

End;

If Not IsCorrect Then

Begin

Path := CheckFileInputPath();

Text := '';

End;

Until IsCorrect;

Writeln('Данные из файла успешно считаны.');

Writeln;

GetTextFromFile := Text;

End;

Function InputText(): AnsiString;

Var

Choice: Integer;

Text: AnsiString;

FInPath: String;

Begin

Choice := CheckChoiceInput

('Если Вы хотите вводить данные в консоль, введите 0. Если использовать файл,

введите 1.');

If Choice = 0 Then

Text := InputTextFromConsole()

Else

Begin

FInPath := CheckFileInputPath();

Text := GetTextFromFile(FInPath);

Writeln('Введенный текст:', #10, Text);

End;

InputText := Text;

End;

Function FormatWord(Count: Integer; Word, ResultText: AnsiString): AnsiString;

Begin

If Odd(Count) Then

ResultText := ResultText + OPENBRACE + Word + CLOSEBRACE

Else

ResultText := ResultText + AnsiUpperCase(Word);

FormatWord := ResultText;

End;

Function FormatText(Text: AnsiString): AnsiString;

Var

I, Count: Integer;

Word, ResultText: AnsiString;

Begin

Word := '';

ResultText := '';

Count := 1;

For I := 1 To Length(Text) Do

If (Text[I] In LETTERS) Or (Text[I] In NUMBERS) Then

Word := Word + Text[I]

Else

If Word <> '' Then

Begin

ResultText := FormatWord(Count, Word, ResultText) + Text[I];

Word := '';

Inc(Count);

End

Else

ResultText := ResultText + Text[I];

If Word <> '' Then

ResultText := FormatWord(Count, Word, ResultText);

FormatText := ResultText;

End;

Function CheckFileOutputPath(): String;

Var

Path: String;

IsCorrect: Boolean;

Begin

Repeat

IsCorrect := True;

Writeln('Введите путь к файлу, в который нужно записать результат.');

Readln(Path);

IsCorrect := IsFilePathCorrect(Path);

If IsCorrect And FileIsReadOnly(Path) Then

Begin

Writeln('Введенный Вами файл доступен только для чтения. Повторите

попытку.');

IsCorrect := False;

End;

Until IsCorrect;

CheckFileOutputPath := Path;

End;

Procedure WriteResultIntoFile(ResultText: AnsiString; Path: String);

Var

FOut: TextFile;

IsCorrect: Boolean;

Begin

Repeat

IsCorrect := True;

Try

Try

AssignFile(FOut, Path);

Rewrite(FOut);

Writeln(FOut, ResultText);

Finally

CloseFile(FOut);

End;

Except

Writeln('Произошла ошибка. Повторите попытку.');

IsCorrect := False;

Path := CheckFileOutputPath();

End;

Until IsCorrect;

Writeln('Результат записан.');

End;

Procedure OutputText(ResultText: AnsiString);

Var

Choice: Integer;

FOutPath: String;

Begin

Choice := CheckChoiceInput

('Если Вы хотите вывести результат в консоль, введите 0. Если в файл, введите

1.');

If Choice = 0 Then

Writeln('Результат:', #10, ResultText)

Else

Begin

FOutPath := CheckFileOutputPath();

WriteResultIntoFile(ResultText, FOutPath);

End;

End;

Var

Text, ResultText: AnsiString;

Begin

WriteCondition();

Text := InputText();

ResultText := FormatText(Text);

OutputText(ResultText);

Readln;

End.

**КОД ПРОГРАММЫ С++:**

#include <iostream>

#include <fstream>

#include <set>

#include <string>

#include <locale>

#include <Windows.h>

using namespace std;

const char OPENBRACE = '(';

const char CLOSEBRACE = ')';

const set<char> NUMBERS {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9'};

set<char> fillLettersSet() {

set<char> LETTERS;

for (char i = 'A'; i < 'Z' + 1; i++)

LETTERS.insert(i);

for (char i = 'a'; i < 'z' + 1; i++)

LETTERS.insert(i);

for (char i = 'А'; i < 'Я' + 1; i++)

LETTERS.insert(i);

for (char i = 'а'; i < 'я' + 1; i++)

LETTERS.insert(i);

LETTERS.insert('Ё');

LETTERS.insert('ё');

return LETTERS;

}

void writeCondition() {

cout << "Данная программа в каждом четном слове текста заменяет все буквы на

прописные,\nа каждое нечетное слово заключает в круглые скобки.\n";

}

int checkChoiceInput(string outputMessage) {

int num;

num = 0;

bool isIncorrect;

do {

isIncorrect = false;

cout << outputMessage;

cin >> num;

if (cin.get() != '\n') {

cin.clear();

while (cin.get() != '\n');

isIncorrect = true;

cout << "Введенные данные не соответствуют условию. Повторите

попытку.\n";

}

if ((!isIncorrect) && (num != 0) && (num != 1)) {

isIncorrect = true;

cout << "Введенные данные не соответствуют условию. Повторите

попытку.\n";

}

} while (isIncorrect);

cout << "\n";

return num;

}

bool isTextIncorrect(string text, const set<char> LETTERS) {

bool isIncorrect;

isIncorrect = true;

if (text == "")

cout << "В тексте нет символов. Повторите попытку.\n";

else {

int i;

i = 0;

while (isIncorrect && (i < text.size() + 1)) {

if (LETTERS.count(text[i]))

isIncorrect = false;

i++;

}

if (isIncorrect)

cout << "В тексте нет букв. Повторите попытку.\n";

}

return isIncorrect;

}

string inputTextFromConsole(const set<char> LETTERS) {

string text;

do {

cout << "Введите текст.\n";

getline(cin, text, '\n');

} while (isTextIncorrect(text, LETTERS));

return text;

}

bool isFileExtIncorrect(string path) {

bool isIncorrect;

isIncorrect = true;

if (path.substr(path.size() - 4) == ".txt")

isIncorrect = false;

else

cout << "Введенный Вами файл не является текстовым. Повторите попытку.\n";

return isIncorrect;

}

string checkFileInputPath() {

string path;

bool isIncorrect;

do {

isIncorrect = false;

cout << "Введите путь к файлу, содержащему текст.\n";

cin >> path;

ifstream fin(path);

fin.open(path);

if (!fin.is\_open()) {

cout << "Введенного файла не существует. Повторите попытку.\n";

isIncorrect = true;

}

if (!isIncorrect)

isIncorrect = isFileExtIncorrect(path);

fin.close();

} while (isIncorrect);

return path;

}

string getTextFromFile(string path, const set<char> LETTERS) {

bool isIncorrect;

string line;

string text;

do {

isIncorrect = false;

ifstream fin(path);

try {

while (!fin.eof()) {

getline(fin, line);

text += line;

}

isIncorrect = isTextIncorrect(text, LETTERS);

}

catch (string errorMessage) {

cout << "Произошла ошибка. Повторите попытку.\n";

isIncorrect = true;

}

fin.close();

if (isIncorrect) {

path = checkFileInputPath();

text = "";

}

} while (isIncorrect);

cout << "Данные из файла успешно считаны.\n\n";

return text;

}

string inputText(const set<char> LETTERS) {

int choice;

string text;

choice = 0;

choice = checkChoiceInput("Если Вы хотите вводить данные в консоль, введите 0.

Если использовать файл, введите 1.\n");

if (choice == 0)

text = inputTextFromConsole(LETTERS);

else {

string finPath = checkFileInputPath();

text = getTextFromFile(finPath, LETTERS);

cout << "Введенный текст:\n" << text << "\n";

}

return text;

}

string formatWord(int count, string word, string resultText) {

locale locRus("Russian");

if (count % 2 != 0)

resultText += OPENBRACE + word + CLOSEBRACE;

else {

for (int j = 0; j < word.size() + 1; j++)

word[j] = toupper(word[j], locRus);

resultText += word;

}

return resultText;

}

string formatText(string text, const set<char> LETTERS) {

int count;

string word;

string resultText;

count = 1;

word = "";

resultText = "";

for (int i = 0; i < text.size() + 1; i++)

if ((LETTERS.count(text[i])) || (NUMBERS.count(text[i])))

word += text[i];

else

if (word != "") {

resultText = formatWord(count, word, resultText) + text[i];

word = "";

count++;

}

else

resultText += text[i];

if (word != "")

resultText = formatWord(count, word, resultText);

return resultText;

}

string checkFileOutputPath() {

string path;

bool isIncorrect;

cout << endl;

do {

isIncorrect = false;

cout << "Введите путь к файлу, в который нужно записать результат.\n";

cin >> path;

ofstream fout(path);

fout.open(path);

if (!fout.is\_open()) {

cout << "Произошла ошибка. Повторите попытку.\n";

isIncorrect = true;

}

if (!isIncorrect)

isIncorrect = isFileExtIncorrect(path);

fout.close();

} while (isIncorrect);

cout << endl;

return path;

}

void writeResultIntoFile(string resultText, string path) {

bool isIncorrect;

do {

isIncorrect = false;

ofstream fout(path);

try {

fout << resultText;

}

catch (string errorMessage) {

cout << "Произошла ошибка. Повторите попытку.\n";

isIncorrect = true;

}

fout.close();

if (isIncorrect)

path = checkFileOutputPath();

} while (isIncorrect);

cout << "Результат записан.";

}

void outputText(string resultText) {

int choice;

choice = 0;

choice = checkChoiceInput("\nЕсли Вы хотите вывести результат в консоль, введите

0. Если в файл, введите 1.\n");

if (choice == 0)

cout << "Результат:\n" << resultText;

else {

string foutPath = checkFileOutputPath();

writeResultIntoFile(resultText, foutPath);

}

}

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

string text;

string resultText;

const set<char> LETTERS = fillLettersSet();

writeCondition();

text = inputText(LETTERS);

resultText = formatText(text, LETTERS);

outputText(resultText);

return 0;

}

**КОД ПРОГРАММЫ JAVA:**

import java.util.Scanner;

import java.util.HashSet;

import java.io.\*;

public class Main {

static final HashSet<Character> NUMBERS = new HashSet<Character>();

static final HashSet<Character> LETTERS = new HashSet<Character>();

static final char OPENBRACE = '(';

static final char CLOSEBRACE = ')';

private static HashSet<Character> fillNumbersSet() {

for (char i = '1'; i < ('9' + 1); i++)

NUMBERS.add(i);

return NUMBERS;

}

private static HashSet<Character> fillLettersSet() {

for (char i = 'A'; i < ('Z' + 1); i++)

LETTERS.add(i);

for (char i = 'a'; i < ('z' + 1); i++)

LETTERS.add(i);

for (char i = 'А'; i < ('Я' + 1); i++)

LETTERS.add(i);

for (char i = 'а'; i < ('я' + 1); i++)

LETTERS.add(i);

LETTERS.add('Ё');

LETTERS.add('ё');

return LETTERS;

}

private static void writeCondition() {

System.out.println("Данная программа в каждом четном слове текста заменяет

все буквы на прописные,\nа каждое нечетное слово

заключает в круглые скобки.");

}

private static int checkChoiceInput(String outputMessage, Scanner scan) {

int num;

num = 0;

boolean isIncorrect;

do {

isIncorrect = false;

System.out.println(outputMessage);

try {

num = Integer.parseInt(scan.nextLine());

}

catch (NumberFormatException e){

System.out.println("Введенные данные не соответствуют условию.

Повторите попытку.");

isIncorrect = true;

}

if (!isIncorrect && (num != 0) && (num != 1)) {

isIncorrect = true;

System.out.println("Введенные данные не соответствуют условию.

Повторите попытку.");

}

} while (isIncorrect);

System.out.println();

return num;

}

private static boolean isTextIncorrect(String text) {

boolean isIncorrect;

isIncorrect = true;

if (text.isEmpty())

System.out.println("В тексте нет символов. Повторите попытку.");

else {

int i = 0;

while (isIncorrect && (i < text.length())) {

if (LETTERS.contains(text.charAt(i)))

isIncorrect = false;

i++;

}

if (isIncorrect)

System.out.println("В тексте нет букв. Повторите попытку.");

}

return isIncorrect;

}

private static String inputTextFromConsole(Scanner scan) {

String text;

do {

System.out.println("Введите текст.");

text = scan.nextLine();

} while (isTextIncorrect(text));

return text;

}

private static boolean isFilePathIncorrect(String path, File file) {

boolean isIncorrect;

isIncorrect = false;

if (!file.exists()) {

System.out.println("Введенного файла не существует. Повторите попытку.");

isIncorrect = true;

}

else

if (!path.endsWith(".txt")) {

System.out.println("Введенный Вами файл не является текстовым.

Повторите попытку.");

isIncorrect = true;

}

return isIncorrect;

}

private static String checkFileInputPath(Scanner scan) {

String path;

File file;

do {

System.out.println("Введите путь к файлу, содержащему текст.");

path = scan.nextLine();

file = new File(path);

} while (isFilePathIncorrect(path, file));

return path;

}

private static String getTextFromFile(String path, Scanner scan) {

boolean isIncorrect;

String line;

String text;

text = "";

do {

isIncorrect = false;

try (Scanner scanFile = new Scanner(new FileReader(path))) {

while (scanFile.hasNext()) {

line = scanFile.nextLine();

text += line;

}

} catch (Exception e){

System.out.println("Данные выбранного файла не соответствуют условию.

Повторите попытку.");

isIncorrect = true;

}

if (!isIncorrect)

isIncorrect = isTextIncorrect(text);

if (isIncorrect) {

path = checkFileInputPath(scan);

text = "";

}

} while (isIncorrect);

System.out.println("Данные из файла успешно считаны.");

System.out.println();

return text;

}

private static String inputText(Scanner scan) {

int choice;

String text;

choice = 0;

choice = checkChoiceInput("Если Вы хотите вводить данные в консоль, введите

0. Если использовать файл, введите 1.", scan);

if (choice == 0)

text = inputTextFromConsole(scan);

else {

String finPath = checkFileInputPath(scan);

text = getTextFromFile(finPath, scan);

System.out.println("Введенный текст:\n" + text);

}

return text;

}

private static String formatWord(int count, String word, String resultText) {

if (count % 2 != 0)

resultText += OPENBRACE + word + CLOSEBRACE;

else {

word = word.toUpperCase();

resultText += word;

}

return resultText;

}

private static String formatText(String text) {

int count;

String word;

String resultText;

count = 1;

word = "";

resultText = "";

for (int i = 0; i < text.length(); i++)

if (LETTERS.contains(text.charAt(i)) || NUMBERS.contains(text.charAt(i)))

word += text.charAt(i);

else

if (word != "") {

resultText = formatWord(count, word, resultText) + text.charAt(i);

word = "";

count++;

}

else

resultText += text.charAt(i);

if (word != "")

resultText = formatWord(count, word, resultText);

return resultText;

}

private static String checkFileOutputPath(Scanner scan) {

String path;

boolean isIncorrect;

do {

isIncorrect = false;

System.out.println("Введите путь к файлу, в который нужно записать

результат.");

path = scan.nextLine();

File file = new File(path);

isIncorrect = isFilePathIncorrect(path, file);

if (!isIncorrect & !file.canWrite()) {

System.out.println("Введенный Вами файл доступен только для чтения.

Повторите попытку.");

isIncorrect = true;

}

} while (isIncorrect);

System.out.println();

return path;

}

private static void writeResultIntoFile(String resultText, String path, Scanner

scan) {

boolean isIncorrect;

do {

isIncorrect = false;

try (FileWriter fw = new FileWriter(new File(path))) {

fw.write(resultText);

} catch (Exception e) {

System.out.println("Произошла ошибка. Повторите попытку.");

isIncorrect = true;

}

if (isIncorrect)

path = checkFileOutputPath(scan);

} while (isIncorrect);

System.out.println("Результат записан.");

}

private static void outputText(String resultText, Scanner scan) {

int choice;

choice = 0;

choice = checkChoiceInput("\nЕсли Вы хотите вывести результат в консоль,

введите 0. Если в файл, введите 1.", scan);

if (choice == 0)

System.out.println("Результат:\n" + resultText);

else {

String foutPath = checkFileOutputPath(scan);

writeResultIntoFile(resultText, foutPath, scan);

}

}

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

String text;

String resultText;

fillNumbersSet();

fillLettersSet();

writeCondition();

text = inputText(scan);

resultText = formatText(text);

outputText(resultText, scan);

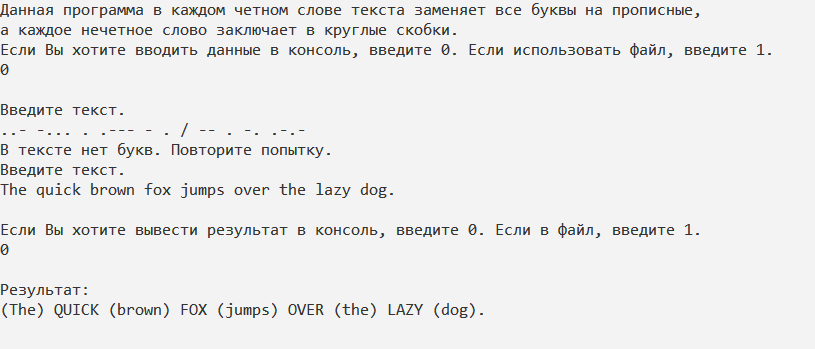
scan.close();

}

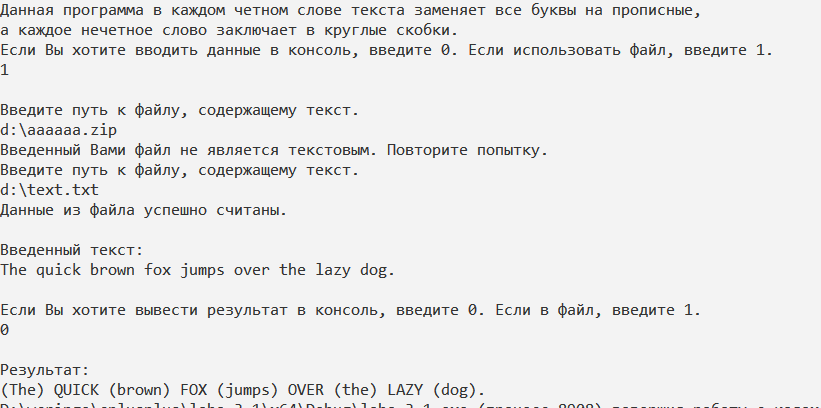
}

**СКРИНШОТЫ:**

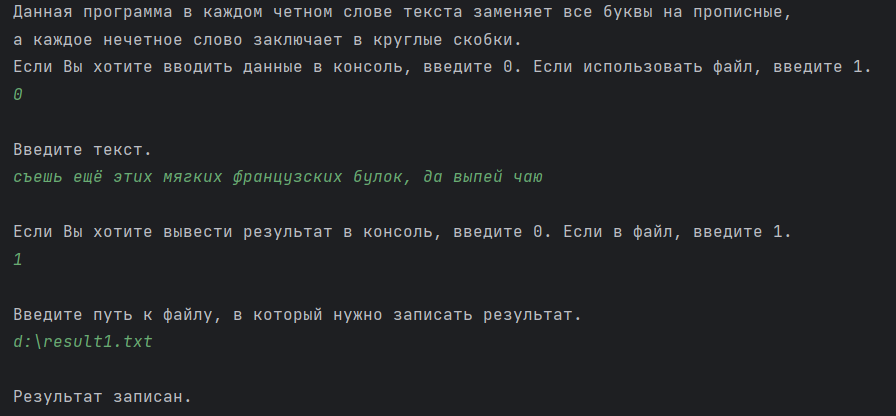
**Delphi:**

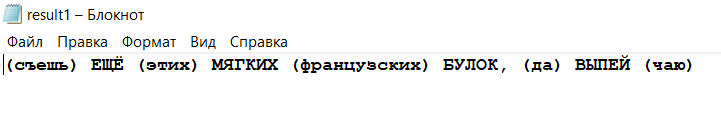


**C++:**



**Java:**





**БЛОК-СХЕМА:**









