УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе №3.1

по предмету «Основы алгоритмизации и программирования»

Вариант 20

Выполнил:

Машевский Д.В

Гр. 351003

Проверил:

Данилова Г. В.

Минск 2023

**Задание:**

Задана строка символов, состоящая из букв, цифр, точек, символов «+» и «-». Выделить подстроку, состоящую из цифр, соответствующую целому числу (т.е. начинается со знака «+» или «-» и внутри подстроки нет букв и точки).

**Код Delphi:**

Program Laba31;

Uses

SysUtils;

Type

TSet = Set of Char;

Const

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

Procedure GetInfo();

Begin

Writeln('This program selects from a string of characters consisting of letters,

numbers, dots, characters "+" and "-," a substring consisting of numbers

corresponding to an integer');

End;

Procedure EnterStr();

Begin

Writeln('Enter your line');

End;

Function ChooseOption(): Integer;

Var

Choise: Integer;

IsCorrect: Boolean;

Begin

Choise := 0;

Writeln('Enter 0 if you want to work with data from the console, and 1 if from

the file:');

Repeat

IsCorrect := True;

Try

Readln(Choise);

Except

Writeln('Invalid data entered. Re-enter:');

IsCorrect := False;

End;

If (Choise <> 0) and (Choise <> 1) then

Begin

IsCorrect := False;

Writeln('Re-enter');

End;

Until IsCorrect;

ChooseOption := Choise;

End;

Function InputStrConsole(): String;

Var

Str: String;

IsCorrect: Boolean;

Begin

EnterStr();

Repeat

IsCorrect := True;

Readln(Str);

If (Length(Str) = 0) Then

Begin

IsCorrect := False;

Writeln('Error. Re-enter');

End;

Until IsCorrect;

InputStrConsole := Str;

End;

Function PTF(): String;

Var

Path: String;

IsCorrect: Boolean;

Begin

Repeat

IsCorrect := True;

Write('Path to file: ');

Readln(Path);

If Not FileExists(Path) Then

Begin

Write('No such file was found. ');

IsCorrect := False;

End;

If IsCorrect And (ExtractFileExt(Path) <> '.txt') Then

Begin

IsCorrect := False;

Write('The file must have a txt extension.');

End;

Until IsCorrect;

PTF := Path;

End;

Function IsCorrectStrFromFile(Var Str: String; Var F: TextFile): Boolean;

Var

IsCorrect: Boolean;

Begin

IsCorrect := True;

If Not Eof(F) Then

Begin

Readln(F, Str);

If (Length(Str) = 0) Then

Begin

Writeln('Verify that the string you entered is correct');

IsCorrect := False;

End;

If (IsCorrect) and Not Eof(F) Then

Begin

Writeln('Remove extra data');

IsCorrect := False;

End;

End

Else

Begin

Writeln('No data for row');

IsCorrect := False;

End;

IsCorrectStrFromFile := IsCorrect;

End;

Function InputStrFromFile(): String;

Var

Str, Path: String;

Fl: TextFile;

IsCorrect: Boolean;

Begin

Repeat

Path := PTF();

AssignFile(Fl, Path);

Try

Try

Reset(Fl);

IsCorrect := IsCorrectStrFromFile(Str, Fl);

Finally

Close(Fl);

End;

Except

IsCorrect := False;

Writeln('Файл нельзя прочитать');

End;

Until IsCorrect;

InputStrFromFile := Str;

End;

Function StartWork(Str: String): String;

Var

IsCorrect: Boolean;

I, N: Integer;

Number: String;

Begin

Number := 'not exist';

IsCorrect := True;

N := Length(Str) + 1;

I := 1;

While I < N Do

Begin

If (IsCorrect And ((Str[I] = '+') Or (Str[I] = '-'))) Then

Begin

Number := Str[I];

Inc(I);

While ((I < N) And (Str[I] In DIGITS)) Do

Begin

Number := Number + Str[I];

Inc(I);

End;

IsCorrect := Length(Number) = 1;

If IsCorrect Then

Number := 'not exist';

End

Else

Inc(I);

End;

StartWork := Number;

End;

Function GiveFResult(Choise: Integer): String;

Var

Str, FinalStr: String;

Begin

If Choise = 0 Then

Begin

Str := InputStrConsole();

End

Else

Begin

Str := InputStrFromFile();

End;

FinalStr := StartWork(Str);

GiveFResult := FinalStr;

End;

Procedure OutputInFile(FinalStr: String);

Var

FileName: String;

MyFile: TextFile;

Begin

AssignFile(MyFile, FileName);

Rewrite(MyFile);

Writeln(MyFile, 'Your extracted substring:', FinalStr);

CloseFile(MyFile);

End;

Procedure OutputInConsole(FinalStr: String);

Begin

Writeln('Your extracted substring:', FinalStr);

End;

Procedure OutputAnswer(FinalStr: String; Choise: Integer);

Begin

If (Choise = 1) Then

OutputInConsole(FinalStr)

Else

OutputInFile(FinalStr);

End;

Var

ChoiseIn, ChoiseOut: Integer;

FinalStr: String;

Begin

GetInfo();

ChoiseIn := ChooseOption();

FinalStr := GiveFResult(ChoiseIn);

ChoiseOut := ChooseOption();

OutputAnswer(FinalStr, ChoiseOut);

Readln;

End.

**Код С++:**

#include <iostream>

#include <set>

#include <string>

#include <fstream>

using namespace std;

void getInfo() {

cout << "This program selects from a string of characters consisting of

letters, numbers, dots, characters + and - a substring consisting

of numbers corresponding to an integer" << endl;

}

void enterLine() {

cout << "Enter your line" << endl;

}

string startWork(string str) {

string finalset;

int i, n;

bool isNumbNotExist;

isNumbNotExist = true;

n = str.length();

i = 0;

finalset = "not exist";

while (i < n) {

if (isNumbNotExist && (str[i] == '+' || str[i] == '-')) {

finalset = str[i];

i++;

while (i < n && isdigit(str[i]))

finalset += str[i++];

isNumbNotExist = finalset.length() == 1;

if (isNumbNotExist)

finalset = "not exist";

}

else

++i;

}

return finalset;

}

int chooseOption() {

int choice;

bool isInCorrect;

cout << "Enter 0 if you want to work with data from the console, and 1 if

from the file:";

do {

isInCorrect = false;

cin >> choice;

if (cin.fail() || cin.peek() != '\n') {

isInCorrect = true;

cout << "Invalid data entered. Re-enter:";

cin.clear();

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

}

if (!(isInCorrect) && (choice != 1) && (choice != 0)) {

isInCorrect = true;

cout << "Re-enter: ";

cin.clear();

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

}

} while (isInCorrect);

return choice;

}

string inputStrConsole() {

string str;

bool isInCorrect;

do {

enterLine();

isInCorrect = false;

cin >> str;

if (str.empty() || str[0] == ' ') {

isInCorrect = true;

cerr << "Re-enter: " << endl;

}

} while (isInCorrect);

return str;

}

bool checkExtension(string path) {

bool isInCorrect = true;

if ((path[path.length() - 1] == 't') &&

(path[path.length() - 2] == 'x') &&

(path[path.length() - 3] == 't') &&

(path[path.length() - 4] == '.')) {

isInCorrect = false;

}

return isInCorrect;

}

string pTF() {

string path;

bool isInCorrect;

do {

isInCorrect = false;

cout << "Path to file: " << endl;

cin >> path;

fstream fal(path);

if (checkExtension && !fal.is\_open()) {

cout << "Re-enter:" << endl;

isInCorrect = true;

}

fal.close();

} while (isInCorrect);

return path;

}

string takeStrFromFile(string path) {

string str;

bool isInCorrect;

ifstream fin(path);

do {

isInCorrect = false;

if (!fin.eof()) {

getline(fin, str);

}

else {

cerr << "No data for row" << endl;

isInCorrect = true;

path = pTF();

}

if (!isInCorrect && !fin.eof()) {

cerr << "Remove extra data" << endl;

isInCorrect = true;

path = pTF();

}

} while (isInCorrect);

fin.close();

return str;

}

string giveFResult(int choise){

string finalset;

string str, path;

if (choise == 0) {

str = inputStrConsole();

}

else {

path = pTF();

str = takeStrFromFile(path);

}

finalset = startWork(str);

return finalset;

}

void printInFile(string path, string finalSet) {

bool isInCorrect;

ofstream fout(path);

do {

isInCorrect = false;

if (finalSet.empty()) {

fout << "No characters needed";

}

else {

for (auto st : finalSet) {

fout << st << '\t';

}

}

} while (isInCorrect);

fout.close();

}

void outputAnswer(string finalSet, int outputChoice) {

string path;

if (outputChoice == 0) {

if (finalSet.empty()) {

cout << "No characters needed";

}

else {

for (auto i = finalSet.begin(); i != finalSet.end(); i++) {

cout << \*i << '\t';

}

}

}

else {

path = pTF();

printInFile(path, finalSet);

}

}

int main() {

getInfo();

string finalSet;

int inChoice, outChoice;

inChoice = chooseOption();

finalSet = giveFResult(inChoice);

outChoice = chooseOption();

outputAnswer(finalSet, outChoice);

return 0;

}

**Код Java:**

import java.util.Scanner;

import java.io.\*;

import java.util.HashSet;

import java.util.Arrays;

public class Main {

static void getInfo() {

System.out.println("This program selects from a string of characters

consisting of letters, numbers, dots, characters + and –

a substring consisting of numbers corresponding to an

integer");

}

static Scanner scan = new Scanner(System.in);

public static String chooseOption() {

boolean isInCorrect;

String choise;

System.out.println("Enter 0 if you want to work with data from the console,

and 1 if from the file:");

do {

isInCorrect = false;

choise = scan.nextLine();

if (!choise.equalsIgnoreCase("0") &&

!choise.equalsIgnoreCase("1")){

System.err.println("Re-enter:");

isInCorrect = true;

}

} while (isInCorrect);

return choise;

}

public static String inputStrConsole() {

String str;

boolean isInCorrect;

do {

System.out.println("Enter your line:");

str = scan.nextLine();

isInCorrect = (str.isEmpty() || str.charAt(0) == ' ');

} while (isInCorrect);

return str;

}

static String getNumFromLine(String line) {

String numb;

int i, size;

boolean isNumbNotExist;

isNumbNotExist = true;

size = line.length();

i = 0;

numb = "not exist";

while (i < size) {

if (isNumbNotExist && (line.charAt(i) == '+' || line.charAt(i) == '-')) {

numb = String.valueOf(line.charAt(i));

i++;

while (i < size && Character.isDigit(line.charAt(i)))

numb += line.charAt(i++);

isNumbNotExist = numb.length() == 1;

if (isNumbNotExist)

numb = "not exist";

}

else

++i;

}

return numb;

}

public static String pTF() {

String path;

boolean isInCorrect;

do {

isInCorrect = false;

System.out.println("Path to file: ");

path = scan.nextLine();

File file = new File(path);

if (!file.exists() | !path.endsWith(".txt") | file.isDirectory()) {

isInCorrect = true;

System.err.println("Check file settings");

}

} while (isInCorrect);

return path;

}

public static String InputStrFromFile() {

String str = "", path;

boolean isInCorrect;

do {

path = pTF();

try (BufferedReader br = new BufferedReader(new FileReader(path))) {

if ((str = br.readLine()) != null) {

isInCorrect = (str.isEmpty() || str.charAt(0) == ' ');

} else {

isInCorrect = true;

System.err.println("There is no data in the file for the row");

}

if (!isInCorrect & br.readLine() != null) {

isInCorrect = true;

System.err.println("Remove extra data");

}

} catch (IOException e) {

System.err.println(e.getMessage());

isInCorrect = true;

}

} while (isInCorrect);

return str;

}

public static void printInFile(String finalSet) {

boolean isInCorrect;

String path;

do {

path = pTF();

isInCorrect = false;

try (PrintWriter pw = new PrintWriter(new FileWriter(path,

false))) {

if (finalSet.isEmpty())

pw.write("No characters needed");

else

pw.write(String.valueOf(finalSet));

} catch (IOException e) {

System.err.println(e.getMessage());

isInCorrect = true;

}

} while (isInCorrect);

}

public static String giveInfo(String choise) {

String str;

String fSet;

str = (choise.equalsIgnoreCase("0")) ? inputStrConsole() :

InputStrFromFile();

fSet = getNumFromLine(str);

return fSet;

}

public static void outputAnswer(String outputChoice, String finalSet) {

if (outputChoice.equalsIgnoreCase("0")) {

if (finalSet.isEmpty())

System.out.println("No characters needed");

else

System.out.println(finalSet);

} else {

printInFile(finalSet);

}

}

public static void main(String[] args) {

getInfo();

String inChoice, outChoice;

String finalSet;

inChoice = chooseOption();

finalSet = giveInfo(inChoice);

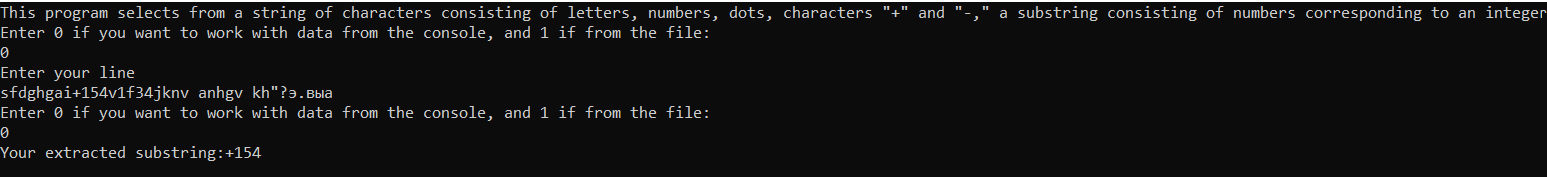
outChoice = chooseOption();

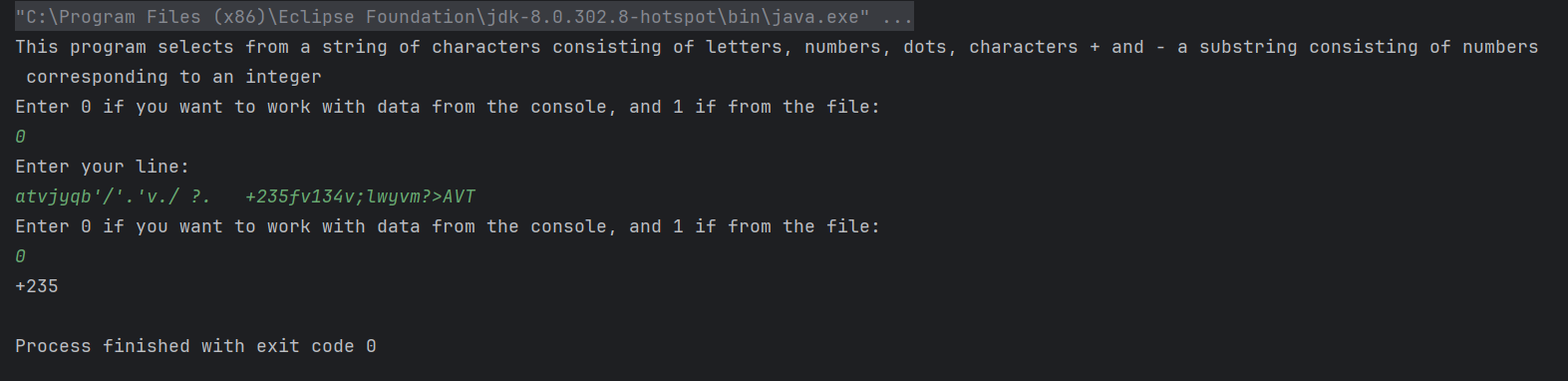
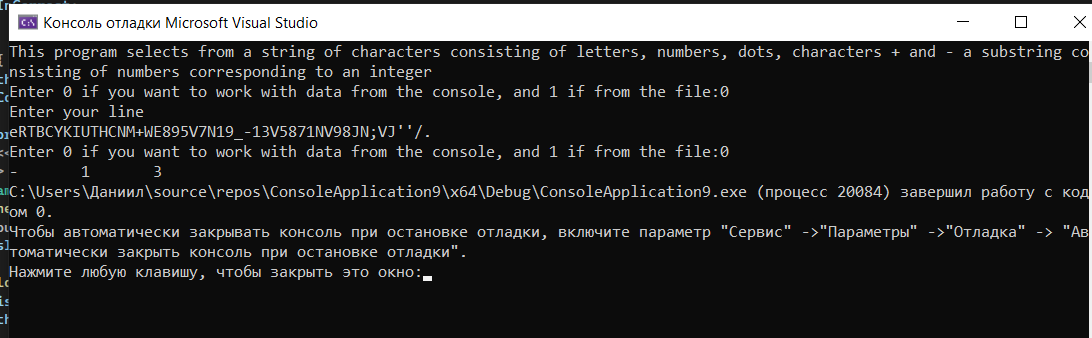
outputAnswer(outChoice, finalSet);

}

}

**Скриншоты:**

****



**Блок-схема:**

