УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе №3.4

по предмету «Основы алгоритмизации и программирования»

Вариант 5

Выполнил:

Бражалович А. И.

Гр. 351005

Проверил:

Данилова Г. В.

Минск 2023

**Задание:**

Перевести число из 10 с/с в 16 с/с.

**Код программы Delphi:**

Program Lab3\_3;

Uses

System.SysUtils;

Type

TMatrix = Array Of Array Of Integer;

TArr = Array Of Integer;

ERRORS\_LIST = (CORRECT, RANGE\_ERR, NUM\_ERR, NOT\_TXT, NOT\_EXIST, NOT\_READABLE, NOT\_WRITEABLE, ORDER\_ERR, CHOICE\_ERR, FILE\_EMPTY);

Const

MIN\_NUMBER = 0;

MAX\_NUMBER = 100000000;

FILE\_CHOICE = 1;

CONSOLE\_CHOICE = 2;

ERRORS: Array [ERRORS\_LIST] Of String = ('', 'Значение не попадает в диапазон!', 'Проверьте корректность ввода данных!', 'Расширение не txt!', 'Проверьте корректность ввода пути к файлу!', 'Файл закрыт для чтения!', 'Файл закрыт для записи!', 'Значения порядков не равны!', 'Проверьте корректность выбора!', 'Файл пуст!');

Procedure PrintTask();

Begin

WriteLn('Данная программа переводит числа из 10 с\с в 16 с\с', #13#10);

End;

Function CheckArea(Num: Integer; MIN, MAX: Integer) : ERRORS\_LIST;

Var

Errors: ERRORS\_LIST;

Begin

Errors := CORRECT;

If (Num < MIN) Or (Num > MAX) Then

Errors := RANGE\_ERR;

CheckArea := Errors;

End;

Procedure PrintError(Error: ERRORS\_LIST);

Begin

WriteLn(ERRORS[Error], #13#10'Повторите попытку: ');

End;

Function CheckNum(MIN, MAX: Integer) : Integer;

Var

Errors: ERRORS\_LIST;

Num: Integer;

Begin

Num := 0;

Repeat

Errors := CORRECT;

Try

Readln(Num);

Except

Errors := CHOICE\_ERR;

End;

If Errors = CORRECT Then

Errors := CheckArea(Num, MIN, MAX);

If Errors <> CORRECT Then

PrintError(Errors);

Until Errors = CORRECT;

CheckNum := Num;

End;

Function CheckInOut() : Boolean;

Var

Num: Integer;

Choose: Boolean;

Begin

Choose := False;

Num := CheckNum(FILE\_CHOICE, CONSOLE\_CHOICE);

If Num = 1 Then

Choose := True;

CheckInOut := Choose;

End;

Function ChooseFileInput() : Boolean;

Var

Choose: Boolean;

Begin

WriteLn('Вы хотите вводить число через файл? (Да - ', 1, ' / Нет - ', 2, ')');

Choose := CheckInOut();

ChooseFileInput := Choose;

End;

Function IsReadable (Var F: TextFile) : ERRORS\_LIST;

Var

Errors: ERRORS\_LIST;

Begin

Errors := CORRECT;

Try

Try

Reset(F);

Finally

CloseFile(F);

End;

Except

Errors := NOT\_READABLE;

End;

IsReadable := Errors;

End;

Procedure FileReading(Var F: TextFile);

Var

Errors: ERRORS\_LIST;

PathToFile: String;

Begin

PathToFile := '';

Repeat

Errors := CORRECT;

Write('Введите путь к файлу с расширением .txt: ');

Readln(PathToFile);

If ExtractFileExt(PathToFile) <> '.txt' Then

Errors := NOT\_TXT;

If Not FileExists(PathToFile) And (Errors = CORRECT) Then

Errors := NOT\_EXIST;

If EOF(F) And (Errors = CORRECT) Then

Errors := FILE\_EMPTY;

If Errors = CORRECT Then

AssignFile(F, PathToFile);

If (Errors = CORRECT) And (IsReadable(F) <> CORRECT) Then

Errors := NOT\_READABLE;

If Errors <> CORRECT Then

PrintError(Errors);

Until Errors = CORRECT;

End;

Procedure FileWriting(Var F: TextFile);

Var

PathToFile: String;

Errors: ERRORS\_LIST;

Begin

PathToFile := '';

Repeat

Errors := CORRECT;

Write('Введите путь к файлу с расширением .txt: ');

Readln(PathToFile);

If ExtractFileExt(PathToFile) <> '.txt' Then

Errors := NOT\_TXT;

If Not FileExists(PathToFile) And (Errors = CORRECT) Then

Errors := NOT\_EXIST;

If Errors = CORRECT Then

AssignFile(F, PathToFile);

If (ERRORS = CORRECT) And (FileIsReadOnly(PathToFile)) Then

Errors := NOT\_WRITEABLE;

If Errors <> CORRECT Then

PrintError(Errors);

Until Errors = CORRECT;

End;

Function ReadDec(): Integer;

Var

RF: TextFile;

Error: ERRORS\_LIST;

DecNum: Integer;

FromFile: Boolean;

Begin

Error := CORRECT;

DecNum := 0;

FromFile := ChooseFileInput();

If FromFile Then

Begin

FileReading(RF);

Reset(RF);

Try

Readln(RF, DecNum);

Except

Error := NUM\_ERR;

End;

If Error = CORRECT Then

Error := CheckArea(DecNum, MIN\_NUMBER, MAX\_NUMBER);

CloseFile(RF);

If Error <> CORRECT Then

PrintError(Error);

End

Else

Begin

Writeln('Введите число [', MIN\_NUMBER, ':', MAX\_NUMBER, ']');

DecNum := CheckNum(MIN\_NUMBER, MAX\_NUMBER)

End;

ReadDec := DecNum;

End;

Function DecToHex (DecNum: Integer): String;

Const

HexCharList: Array [0..15] Of Char = ('0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F');

Var

HexNum: String;

ModFromDec: Integer;

Begin

ModFromDec := 0;

HexNum := '';

While (DecNum > 0) Do

Begin

ModFromDec := DecNum Mod 16;

HexNum := HexNum + HexCharList[ModFromDec];

DecNum := DecNum Div 16;

End;

DecToHex := HexNum;

End;

Function ChooseFileOutput() : Boolean;

Var

Choose: Boolean;

Begin

WriteLn;

WriteLn('Вы хотите выводить ответ через файл? (Да - ', 1, ' / Нет - ', 2, ')');

Choose := CheckInOut();

ChooseFileOutput := Choose;

End;

Procedure PrintResult(HexNum: String);

Var

F: TextFile;

PrintToFile: Boolean;

Begin

PrintToFile := ChooseFileOutput();

If PrintToFile Then

Begin

FileWriting(F);

Append(F);

Write(F, 'Число в 16с/c: ');

Write(F, HexNum);

CloseFile(F);

End

Else

Begin

Write('Число в 16с/c: ');

Write(HexNum);

End;

End;

Var

DecNum: Integer;

HexNum: String;

Begin

PrintTask();

DecNum := ReadDec();

HexNum := DecToHex(DecNum);

PrintResult(HexNum);

ReadLn;

End.

**Код программы С++:**

#include <iostream>

#include <string>

#include <fstream>

using namespace std;

enum ERRORS\_LIST {

CORRECT, RANGE\_ERR, NUM\_ERR, NOT\_TXT, NOT\_EXIST, NOT\_READABLE, NOT\_WRITEABLE, FILE\_EMPTY

};

const string ERRORS[] = {

"", "Значение не попадает в диапазон!", "Проверьте корректность ввода данных!", "Расширение не txt!", "Проверьте корректность ввода пути к файлу!", "Файл закрыт для чтения!", "Файл закрыт для записи!", "Файл пуст!"

};

constexpr int MIN\_NUMBER = 0;

constexpr int MAX\_NUMBER = 1000000;

void printTask()

{

cout << "Данная программа переводит числа из 10 с/с в 16 с/с \n";

}

ERRORS\_LIST checkArea(int num, const int MIN, const int MAX)

{

ERRORS\_LIST error;

error = CORRECT;

if (num < MIN || num > MAX)

error = RANGE\_ERR;

return error;

}

void printError(ERRORS\_LIST error)

{

cout << ERRORS[error] << "\nПовторите попытку: ";

}

int checkNum(int MIN, int MAX)

{

ERRORS\_LIST error;

int num;

num = 0;

do

{

error = CORRECT;

cin >> num;

if (cin.fail())

{

error = NUM\_ERR;

cin.clear();

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

}

if (error == CORRECT && cin.get() != '\n')

{

error = NUM\_ERR;

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

}

if (error == CORRECT)

error = checkArea(num, MIN, MAX);

if (error != CORRECT)

printError(error);

} while (error != CORRECT);

return num;

}

bool checkInOut()

{

const int FILE\_CHOICE = 1;

const int CONSOLE\_CHOICE = 2;

int num;

bool choose;

choose = false;

num = checkNum(FILE\_CHOICE, CONSOLE\_CHOICE);

if (num == 1)

choose = true;

return choose;

}

bool chooseFileInput()

{

bool choose;

choose = false;

cout << "Вы хотите вводить число через файл? (Да - " << 1 << " / Нет - " << 2 << ")\n";

choose = checkInOut();

return choose;

}

void fileReading(string& pathToFile)

{

ERRORS\_LIST error;

do

{

error = CORRECT;

cout << "Введите путь к файлу с расширением .txt: ";

getline(cin, pathToFile);

if (pathToFile.substr(pathToFile.length() - 4, 4) != ".txt")

error = NOT\_TXT;

else if (!ifstream(pathToFile))

error = NOT\_EXIST;

else

{

ifstream file(pathToFile);

if (!file.is\_open())

error = NOT\_READABLE;

else if (file.peek() == ifstream::traits\_type::eof())

{

error = FILE\_EMPTY;

file.close();

}

file.close();

}

if (error != CORRECT)

printError(error);

} while (error != CORRECT);

}

void fileWriting(string& pathToFile)

{

ERRORS\_LIST error;

do

{

error = CORRECT;

cout << "Введите путь к файлу с расширением .txt: ";

getline(cin, pathToFile);

if (pathToFile.substr(pathToFile.length() - 4, 4) != ".txt")

error = NOT\_TXT;

else if (!ifstream(pathToFile))

error = NOT\_EXIST;

else

{

ifstream file(pathToFile);

if (!file.is\_open())

error = NOT\_WRITEABLE;

}

if (error != CORRECT)

printError(error);

} while (error != CORRECT);

}

int readDec()

{

bool fromFile;

int decNum;

ERRORS\_LIST error;

string pathToFile;

decNum = 0;

error = CORRECT;

pathToFile = "";

fromFile = chooseFileInput();

if (fromFile)

{

do

{

fileReading(pathToFile);

ifstream file(pathToFile);

file >> decNum;

if (file.fail())

{

error = NUM\_ERR;

file.clear();

}

if (error == CORRECT)

error = checkArea(decNum, MIN\_NUMBER, MAX\_NUMBER);

file.close();

if (error != CORRECT)

printError(error);

} while (error != CORRECT);

}

else

{

cout << "Введите число [" << MIN\_NUMBER << ":" << MAX\_NUMBER << "] ";

decNum = checkNum(MIN\_NUMBER, MAX\_NUMBER);

}

return decNum;

}

string decToHex(int decNum)

{

string hexNum;

int modFromDec;

const char hexCharList[16] = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F'};

hexNum = "";

modFromDec = 0;

while (decNum > 0)

{

modFromDec = decNum % 16;

hexNum += hexCharList[modFromDec];

decNum /= 16;

}

return hexNum;

}

bool chooseFileOutput()

{

bool choose;

cout << "\nВы хотите выводить ответ через файл? (Да - " << 1 << " / Нет - " << 2 << ")\n";

choose = checkInOut();

return choose;

}

void printResult(string hexNum)

{

string pathToFile;

bool printToFile;

printToFile = chooseFileOutput();

pathToFile = ' ';

if (printToFile)

{

fileWriting(pathToFile);

ofstream fileOut(pathToFile, std::ios::app);

fileOut << "Число в 16с/c: ";

fileOut << hexNum;

fileOut.close();

}

else

{

cout << "Число в 16с/c: ";

cout << hexNum;

}

}

int main()

{

int decNum;

string hexNum;

setlocale(LC\_ALL, "RU");

printTask();

decNum = readDec();

hexNum = decToHex(decNum);

printResult(hexNum);

return 0;

}

**Код программы Java:**

import java.io.\*;

import java.util.Scanner;

public class Main {

public static final int MIN\_NUMBER = 0;

public static final int MAX\_NUMBER = 1000000;

public enum ERRORS\_LIST {

CORRECT, RANGE\_ERR, NUM\_ERR, NOT\_TXT, NOT\_EXIST, NOT\_READABLE, NOT\_WRITEABLE, FILE\_EMPTY

}

public static final String[]

ERRORS = {

"", "Значение не попадает в диапазон!", "Проверьте корректность ввода данных!", "Расширение не txt!", "Проверьте корректность ввода пути к файлу!", "Файл закрыт для чтения!", "Файл закрыт для записи!", "Файл пуст!" };

public static Scanner scanConsole = new Scanner(System.in);

public static File file;

public static void printTask() {

System.out.println("Данная программа переводит числа из 10 с/с в 16 с/с: \n");

}

public static void printError (ERRORS\_LIST error) {

System.out.println(ERRORS[error.ordinal()] + "\nПовторите попытку: ");

}

public static ERRORS\_LIST checkArea(int num, final int MIN, final int MAX) {

ERRORS\_LIST error;

error = ERRORS\_LIST.CORRECT;

if (num < MIN || num > MAX)

error = ERRORS\_LIST.RANGE\_ERR;

return error;

}

public static int checkNum(int MIN, int MAX) {

ERRORS\_LIST error;

int num;

num= 0;

do {

error = ERRORS\_LIST.CORRECT;

try {

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

} catch (NumberFormatException e) {

error = ERRORS\_LIST.NUM\_ERR;

}

if (error == ERRORS\_LIST.CORRECT)

error = checkArea(num, MIN, MAX);

if (error != ERRORS\_LIST.CORRECT)

printError(error);

} while (error != ERRORS\_LIST.CORRECT);

return num;

}

public static boolean checkInOut() {

final int FILE\_CHOICE = 1;

final int CONSOLE\_CHOICE = 2;

int num;

boolean choose;

choose = false;

num = checkNum(FILE\_CHOICE, CONSOLE\_CHOICE);

if (num == 1)

choose = true;

return choose;

}

public static boolean chooseFileInput() {

boolean choose;

choose= true;

System.out.println("Вы хотите вводить число через файл? (Да - " + 1 + " / Нет - " + 2 + ")");

choose = checkInOut();

return choose;

}

public static String ReadPath(){

ERRORS\_LIST error;

String pathToFile;

pathToFile = " ";

scanConsole = new Scanner(System.in);

do {

error = ERRORS\_LIST.CORRECT;

System.out.print("Введите путь к файлу с расширением .txt: ");

pathToFile = scanConsole.nextLine();

if (!pathToFile.endsWith(".txt"))

error = ERRORS\_LIST.NOT\_TXT;

if (error != ERRORS\_LIST.CORRECT) {

printError(error);

}

} while (error != ERRORS\_LIST.CORRECT);

return pathToFile;

}

public static void fileReading () {

ERRORS\_LIST error;

String pathToFile;

pathToFile = "";

do {

pathToFile = ReadPath();

error = ERRORS\_LIST.CORRECT;

file = new File(pathToFile);

if (!(file.exists()))

error = ERRORS\_LIST.NOT\_EXIST;

else {

if (!file.canRead())

error = ERRORS\_LIST.NOT\_READABLE;

else {

try (Scanner fileScanner = new Scanner(file)){

if (!fileScanner.hasNext())

error = ERRORS\_LIST.FILE\_EMPTY;

} catch (FileNotFoundException e) {

printError(error);

}

}

}

if (error != ERRORS\_LIST.CORRECT)

printError(error);

} while (error != ERRORS\_LIST.CORRECT);

}

public static String fileWriting() {

ERRORS\_LIST error;

String pathToFile;

pathToFile = "";

do {

pathToFile = ReadPath();

error = ERRORS\_LIST.CORRECT;

if (!(new File(pathToFile)).exists())

error = ERRORS\_LIST.NOT\_EXIST;

else {

File file = new File(pathToFile);

if (!file.canWrite())

error = ERRORS\_LIST.NOT\_WRITEABLE;

}

if (error != ERRORS\_LIST.CORRECT)

printError(error);

} while (error != ERRORS\_LIST.CORRECT);

return pathToFile;

}

public static int readDecFile() {

int decNum;

decNum = 0;

ERRORS\_LIST error;

error = ERRORS\_LIST.CORRECT;

do {

fileReading();

try (Scanner scanFile = new Scanner(file)) {

decNum = scanFile.nextInt();

error = checkArea(decNum, MIN\_NUMBER, MAX\_NUMBER);

}

catch (Exception e) {

error = ERRORS\_LIST.NUM\_ERR;

}

if (error != ERRORS\_LIST.CORRECT)

printError(error);

} while (error != ERRORS\_LIST.CORRECT);

return decNum;

}

public static int readDec() {

boolean fromFile;

int decNum;

decNum = 0;

fromFile = chooseFileInput();

if (fromFile)

decNum = readDecFile();

else {

System.out.print("Введите число [" + MIN\_NUMBER + ":" + MAX\_NUMBER + "] ");

decNum = checkNum(MIN\_NUMBER, MAX\_NUMBER);

}

return decNum;

}

public static String decToHex(int decNum) {

String hexNum;

int modFromDec;

final char[] hexCharList = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F'};

hexNum = "";

modFromDec = 0;

while (decNum > 0) {

modFromDec = decNum % 16;

hexNum += hexCharList[modFromDec];

decNum /= 16;

}

return hexNum;

}

public static boolean chooseFileOutput() {

boolean choose;

System.out.println("Вы хотите выводить ответ через файл? (Да - " + 1 + " / Нет - " + 2 + ")");

choose = checkInOut();

return choose;

}

public static void printResult(String hexNum) {

String pathToFile;

boolean printToFile;

ERRORS\_LIST error;

printToFile = chooseFileOutput();

pathToFile = "";

error = ERRORS\_LIST.CORRECT;

if (printToFile) {

pathToFile = fileWriting();

try (FileWriter fileWriter = new FileWriter(pathToFile, true)){

fileWriter.write("Число в 16с/c: ");

fileWriter.write(hexNum);

} catch (IOException e) {

error = ERRORS\_LIST.NOT\_WRITEABLE;

printError(error);

};

} else {

System.out.print("Число в 16с/c: ");

System.out.print(hexNum);

}

}

public static void main(String[] args) {

int decNum;

String hexNum;

printTask();

decNum = readDec();

hexNum = decToHex(decNum);

printResult(hexNum);

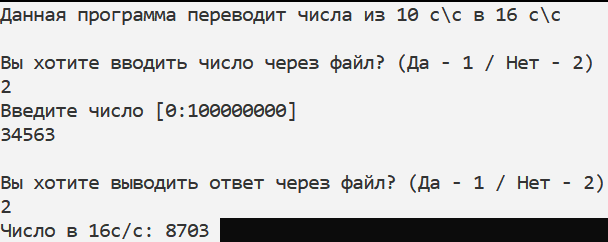
scanConsole.close();

}

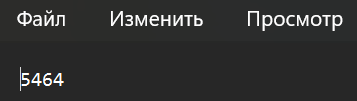
}

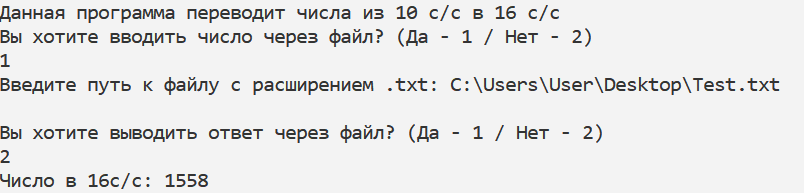
**Скриншоты:**

**Delphi:**

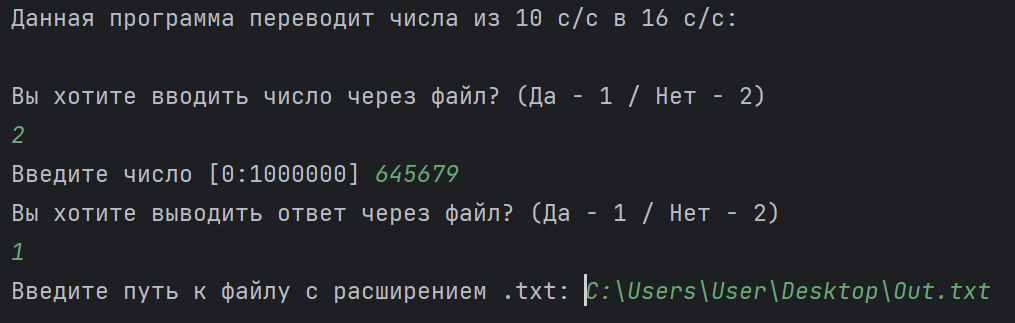
****

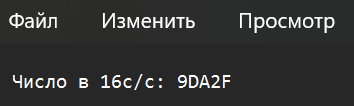
**C++:**

****

****

**Java:**

****

****

**Блок-схема:**



