УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе №3.2

по предмету «Основы алгоритмизации и программирования»

Вариант 15

Выполнила:

Городко К. Е.

Гр. 351005

Проверила:

Данилова Г. В.

Минск 2023

**ЗАДАНИЕ:**

Написать программу, которая формирует множество Y={X1 U X2} и выделить из него подмножество Y1 чисел, которые являются четными

числами, кратными 3. Дано X1={1,2,3,4,5,6}; X2={1,3,4,5,6}.

**КОД ПРОГРАММЫ DELPHI:**

Program LAB3\_2;

Uses

System.SysUtils;

Type

TSet = Set Of Byte;

Const

X1 = [1, 2, 3, 4, 5, 6];

X2 = [1, 3, 4, 5, 6];

BYTEMAX = 255;

Procedure OutputSet(SetOfNum: TSet; SetName: String);

Var

I: Byte;

Begin

Write(SetName, '{ ');

For I := 0 To BYTEMAX Do

If (I In SetOfNum) Then

Write(I, ' ');

Writeln('}');

End;

Function FindSumOfSets(): TSet;

Var

Y: TSet;

Begin

Y := X1 + X2;

FindSumOfSets := Y;

End;

Function FindSubSet(Y: TSet): TSet;

Var

I: Byte;

Y1: TSet;

Begin

Y1 := [];

For I := 0 To BYTEMAX Do

If (I In Y) And (Not Odd(I)) And (I Mod 3 = 0) Then

Y1 := Y1 + [I];

FindSubSet := Y1;

End;

Procedure WriteCondition();

Begin

Writeln('Данная программа формирует объединение двух множеств, и выделяет из него

подмножество четных чисел, кратных 3.');

Writeln('Даны множества:');

OutputSet(X1, 'X1: ');

OutputSet(X2, 'X2: ');

End;

Function CheckChoiceInput(): Integer;

Var

Num: Integer;

IsCorrect: Boolean;

Begin

Repeat

IsCorrect := True;

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

введите 1.');

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);

CheckChoiceInput := Num;

End;

Function CheckFilePath(): String;

Var

Path: String;

IsCorrect: Boolean;

Begin

Repeat

IsCorrect := True;

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

Readln(Path);

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

Else

If FileIsReadOnly(Path) Then

Begin

Writeln('Введенный Вами файл доступен только для чтения.

Повторите попытку.');

IsCorrect := False;

End;

Until IsCorrect;

CheckFilePath := Path;

End;

Procedure WriteSetIntoFile(SetOfNum: TSet; SetName: String; Var FOut: TextFile);

Var

I: Byte;

Begin

Write(FOut, SetName, '{ ');

For I := 0 To BYTEMAX Do

If (I In SetOfNum) Then

Write(FOut, I, ' ');

Writeln(FOut, '}');

End;

Procedure WriteResultIntoFile(Y, Y1: TSet; Path: String);

Var

I: Byte;

IsCorrect: Boolean;

FOut: TextFile;

Begin

Repeat

IsCorrect := True;

Assign(FOut, Path);

Try

Try

Rewrite(FOut);

WriteSetIntoFile(Y, 'Полученное объединение множеств: ', FOut);

WriteSetIntoFile(Y1, 'Полученное подмножество: ', FOut);

Finally

CloseFile(FOut);

End;

Except

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

IsCorrect := False;

Path := CheckFilePath();

End;

Until IsCorrect;

Writeln('Результат записан.');

End;

Var

Choice: Integer;

FOutPath: String;

Y, Y1: TSet;

Begin

WriteCondition();

Y := FindSumOfSets();

Y1 := FindSubSet(Y);

Choice := CheckChoiceInput();

If Choice = 0 Then

Begin

OutputSet(Y, 'Полученное объединение множеств: ');

OutputSet(Y1, 'Полученное подмножество: ');

End

Else

Begin

FOutPath := CheckFilePath();

WriteResultIntoFile(Y, Y1, FOutPath);

End;

End.

**КОД ПРОГРАММЫ С++:**

#include <iostream>

#include <fstream>

#include <set>

using namespace std;

const set<int> X1 {1, 2, 3, 4, 5, 6};

const set<int> X2 {1, 3, 4, 5, 6};

void outputSet(set<int> setOfNum, string setName) {

cout << setName << "{ ";

for (int i : setOfNum)

cout << i << " ";

cout << "}" << endl;

}

set<int> findSumOfSets() {

set<int> y;

y.insert(X1.begin(), X1.end());

y.insert(X2.begin(), X2.end());

return y;

}

set<int> findSubSet(set<int> y) {

set<int> y1;

for (int i : y)

if ((i % 2 == 0) && (i % 3 == 0))

y1.insert(i);

return y1;

}

void writeCondition() {

cout << "Данная программа формирует объединение двух множеств, и выделяет из него

подмножество четных чисел, кратных 3." << endl;

cout << "Даны множества:" << endl;

outputSet(X1, "X1: ");

outputSet(X2, "X2: ");

}

int checkChoiceInput() {

int num;

num = 0;

bool isIncorrect;

do {

isIncorrect = false;

cout << "Если Вы хотите вывести результат в консоль, введите 0. Если в файл,

введите 1." << endl;

cin >> num;

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

cin.clear();

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

isIncorrect = true;

cout << "Введенные данные не соответствуют условию. Повторите попытку."

<< endl;

}

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

isIncorrect = true;

cout << "Введенные данные не соответствуют условию. Повторите попытку."

<< endl;

}

} while (isIncorrect);

return num;

}

string checkFilePath() {

string path;

bool isIncorrect;

do {

isIncorrect = false;

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

cin >> path;

ofstream fout(path);

fout.open(path);

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

cout << "Произошла ошибка. Повторите попытку." << endl;

isIncorrect = true;

}

if ((!isIncorrect) && (path.substr(path.size() - 4) != ".txt")) {

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

<< endl;

isIncorrect = true;

}

fout.close();

} while (isIncorrect);

return path;

}

void writeSetIntoFile(set<int> setOfNum, string setName, ofstream& fout) {

fout << setName << "{ ";

for (int i : setOfNum)

fout << i << " ";

fout << "}" << endl;

}

void writeResultIntoFile(set<int> y, set<int> y1, string path) {

bool isIncorrect;

do {

isIncorrect = false;

ofstream fout(path);

try {

writeSetIntoFile(y, "Полученное объединение множеств: ", fout);

writeSetIntoFile(y1, "Полученное подмножество: ", fout);

}

catch (string errorMessage) {

cout << "Произошла ошибка. Повторите попытку." << endl;

isIncorrect = true;

}

fout.close();

if (isIncorrect)

path = checkFilePath();

} while (isIncorrect);

cout << "Результат записан.";

}

int main() {

setlocale(LC\_ALL, "Russian");

int choice;

choice = 0;

string foutPath;

set<int> y;

set<int> y1;

writeCondition();

y = findSumOfSets();

y1 = findSubSet(y);

choice = checkChoiceInput();

if (choice == 0) {

outputSet(y, "Полученное объединение множеств: ");

outputSet(y1, "Полученное подмножество: ");

}

else {

foutPath = checkFilePath();

writeResultIntoFile(y, y1, foutPath);

}

return 0;

}

**КОД ПРОГРАММЫ JAVA:**

import java.util.Scanner;

import java.util.HashSet;

import java.io.\*;

public class Main {

static final HashSet<Integer> X1 = new HashSet<Integer>();

static final HashSet<Integer> X2 = new HashSet<Integer>();

static final int SETMAXELEMENT = 7;

private static HashSet<Integer> addX1Elements(){

for (int i = 1; i < SETMAXELEMENT; i++)

X1.add(i);

return X1;

}

private static HashSet<Integer> addX2Elements(){

X2.add(1);

for (int i = 3; i < SETMAXELEMENT; i++)

X2.add(i);

return X2;

}

private static void outputSet(HashSet<Integer> set, String setName) {

System.out.print(setName + "{ ");

for (int i : set)

System.out.print(i + " ");

System.out.println("}");

}

private static HashSet<Integer> findSumOfSets() {

HashSet<Integer> y = new HashSet<Integer>();

for (int i = 1; i < SETMAXELEMENT; i++)

if (X1.contains(i) || X2.contains(i))

y.add(i);

return y;

}

private static HashSet<Integer> findSubSet(HashSet<Integer> y) {

HashSet<Integer> y1 = new HashSet<Integer>();

for (int i : y)

if ((i % 2 == 0) && (i % 3 == 0))

y1.add(i);

return y1;

}

private static void writeCondition() {

System.out.println("Данная программа формирует объединение двух множеств, и

выделяет из него подмножество четных чисел, кратных 3.");

System.out.println("Даны множества:");

outputSet(X1, "X1: ");

outputSet(X2, "X2: ");

}

private static int checkChoiceInput(Scanner scan) {

int num;

num = 0;

boolean isIncorrect;

do {

isIncorrect = false;

System.out.println("Если Вы хотите вывести результат в консоль, введите

0. Если в файл, введите 1.");

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);

return num;

}

private static String checkFilePath(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);

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;

}

else

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

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

чтения. Повторите попытку.");

isIncorrect = true;

}

} while (isIncorrect);

return path;

}

private static void writeSetIntoFile(HashSet<Integer> set, String setName,

FileWriter fw) throws IOException {

fw.write(setName + "{ ");

for (int i : set)

fw.write(i + " ");

fw.write("}" + "\n");

}

private static void writeResultIntoFile(HashSet<Integer> y, HashSet<Integer> y1,

String path, Scanner scan) {

boolean isIncorrect;

do {

isIncorrect = false;

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

writeSetIntoFIle(y, "Полученное объединение множеств: ", fw);

writeSetIntoFIle(y1, "Полученное подмножество: ", fw);

} catch (Exception e) {

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

isIncorrect = true;

}

if (isIncorrect)

path = checkFilePath(scan);

} while (isIncorrect);

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

}

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

int choice;

choice = 0;

String foutPath;

HashSet<Integer> y = new HashSet<Integer>();

HashSet<Integer> y1 = new HashSet<Integer>();

addX1Elements();

addX2Elements();

writeCondition();

y = findSumOfSets();

y1 = findSubSet(y);

choice = checkChoiceInput(scan);

if (choice == 0) {

outputSet(y, "Полученное объединение множеств: ");

outputSet(y1, "Полученное подмножество: ");

}

else {

foutPath = checkFilePath(scan);

writeResultIntoFile(y, y1, foutPath, scan);

}

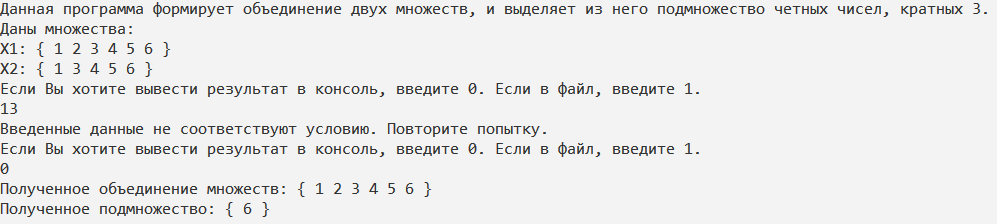
scan.close();

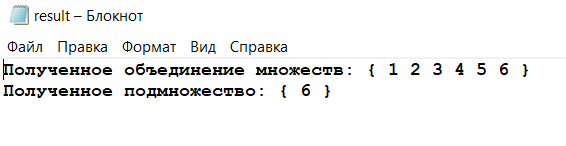
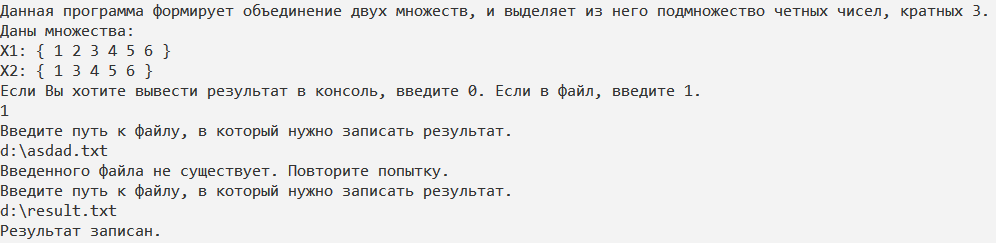
}

}

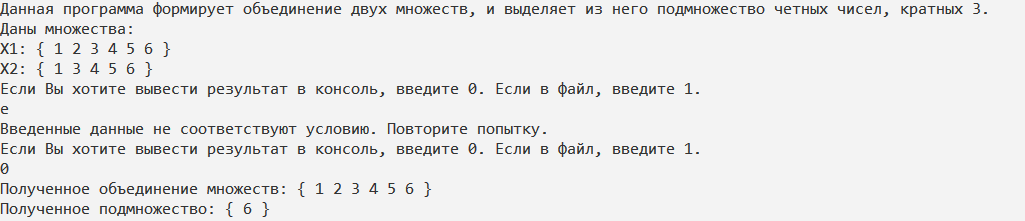
**СКРИНШОТЫ:**

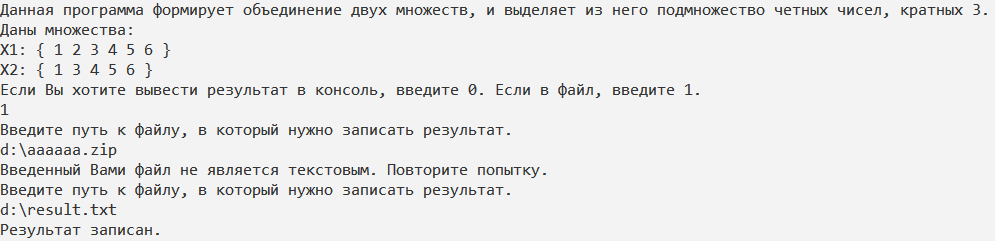
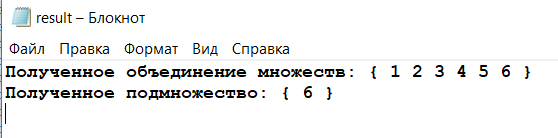
**Delphi:**



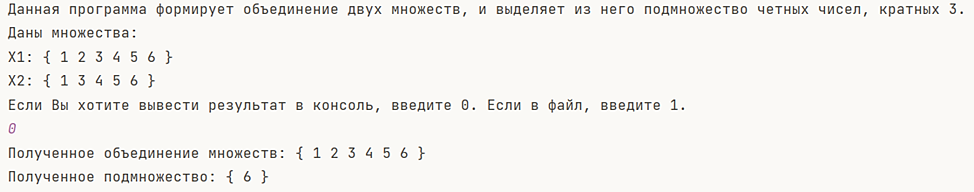


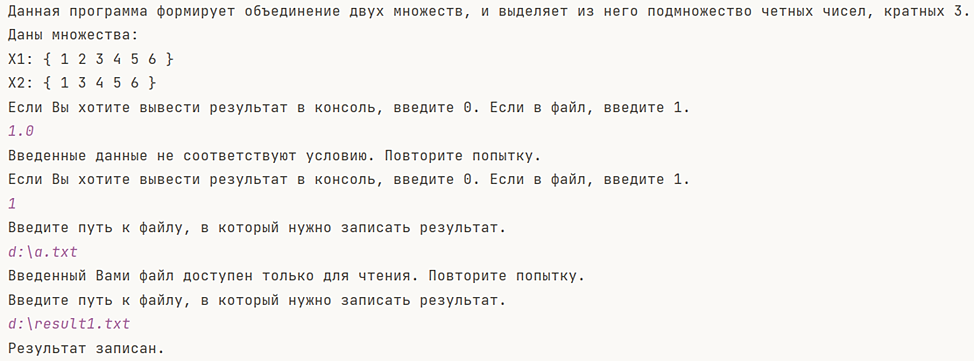
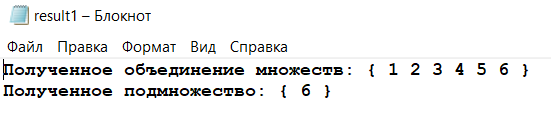
**C++:**



**Java:**



**БЛОК-СХЕМА:**





