Федеральное агентство связи

Ордена Трудового Красного Знамени федеральное государственное

бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

Практическая работа №3

По дисциплине

«Технологии программирования»

Выполнил

студент группы: БСТ1702

Якунин А.В.

Москва

2020

**Цель работы:** изучение основных принципов работы языка программирования Java и выполнение практических задач на нем.

**1 Задача:** Квадратное уравнение ax2 + bx + c = 0 имеет либо 0, либо 1, либо 2 различных решения для действительных значений x. учитывая a, b и c, вы должны вернуть число решений в уравнение.

Решение:

Вызов

System.out.println("Введите три значения квадратного уравнения");

int n11 = sca.nextInt();

int n21 = sca.nextInt();

int n31 = sca.nextInt();

int res1 = solutions(n11, n21,n31);

System.out.println("Число корней = " + res1);

Функция

public static Integer solutions(int a, int b, int c){

int D = (b\*b) - 4 \* a \* c;

if(D > 0){

return 2;

}else if(D == 0){

return 1;

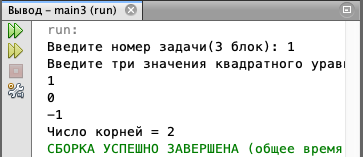
}else{

return 0;

}

}

Результат:



**2 Задача:** Напишите функцию, которая возвращает позицию второго вхождения " zip " в строку, или -1, если оно не происходит по крайней мере дважды. Ваш код должен быть достаточно общим, чтобы передать все возможные случаи, когда "zip" может произойти в строке.

Решение:

Вызов

System.out.print("Введите строку: ");

String st2 = sca.nextLine();

int res2 = findZip(st2);

System.out.println("Второе вхождение zip на " + res2 + " символе");

Функция

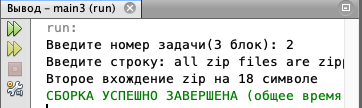
public static int findZip(String st2){

int z = st2.indexOf("zip", st2.indexOf("zip") + 1);

return z;

}

Результат:



**3 Задача:** Создайте функцию, которая проверяет, является ли целое число совершенным числом или нет. Совершенное число - это число, которое можно записать как сумму его множителей, исключая само число.

Например, 6 - это идеальное число, так как 1 + 2 + 3 = 6, где 1, 2 и 3-Все коэффициенты 6. Точно так же 28-это совершенное число, так как 1 + 2 + 4 + 7 + 14 = 28.

Решение:

Вызов

System.out.print("Введите число: ");

int n3 = sca.nextInt();

String res3 = checkPerfect(n3);

System.out.println("Результат: " + res3);

Функция

public static String checkPerfect(int n3)

{

int res3 = 0 ;

for (int i = 1; i< n3; i++){

if (n3 % i == 0 ){

res3 += i;

}

}

if (res3 == n3){

return "true";

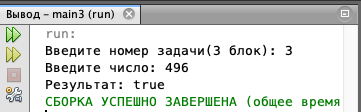
} else {

return "false";

}

}

Результат:



**4 Задача:** Создайте функцию, которая принимает строку и возвращает новую строку с заменой ее первого и последнего символов, за исключением трех условий:

– Если длина строки меньше двух, верните "несовместимо".".

– Если первый и последний символы совпадают, верните "два-это пара.".

Решение:

Вызов

String st4 = new String();

System.out.println("Введите строку:");

st4 = sca.nextLine();

String res4 = flipEndChars(st4);

System.out.println(res4);

Функция

public static String flipEndChars (String st4){

String res4 = "";

if (st4.length() < 2){

res4 = "Incompatible.";

}

else if (st4.charAt(0) == st4.charAt(st4.length() - 1))

{

res4 = "Two's a pair.";

}

else

{

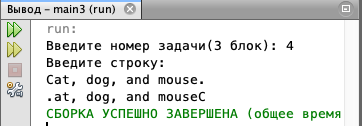
return st4.charAt(st4.length()-1) + st4.substring(1, st4.length()-1) + st4.charAt(0);

}

return res4;

}

Результат:

****

**5 Задача:** Создайте функцию, которая определяет, является ли строка допустимым шестнадцатеричным кодом.

Шестнадцатеричный код должен начинаться с фунтового ключа # и иметь длину ровно 6 символов. Каждый символ должен быть цифрой от 0-9 или буквенным символом от A-F. все буквенные символы могут быть прописными или строчными.

Решение:

Вызов

System.out.print("Введите строку: ");

String st5 = sca.nextLine();

String res5 = isValidHexCode(st5);

System.out.println("Результат: " + res5);

Функция

public static String isValidHexCode(String st5){

int sum = 0;

String res5;

if (st5.charAt(0) == '#' && st5.length() == 7){

for (int i = 1; i <= 6 ; i++){

int a5 = ((int) st5.charAt(i));

if (((a5 >= 48) && (a5 <= 57)) ||((a5 >= 65) && (a5 <= 70)) ||((a5 >= 97) && (a5 <= 102)) ){

sum +=1;

}

}

}else{

res5 = "false";

}

if(sum == 6){

res5 = "true";

}else{

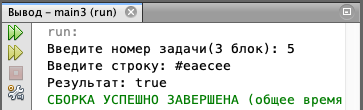
res5 = "false";

}

return res5;

}

Результат:



**6 Задача:** Напишите функцию, которая возвращает true, если два массива имеют одинаковое количество уникальных элементов, и false в противном случае.

Решение:

Вызов

double [] x = new double[]{};

System.out.print("Введите длину 1 массива: ");

int s1 = sca.nextInt();

x = crarr(s1);

double [] y = new double[]{};

System.out.print("Введите длину 2 массива: ");

int s2 = sca.nextInt();

y = crarr(s2);

int x1 = same(x);

int y1 = same(y);

if (x1 == y1){

System.out.println("Результат: true");

}else{

System.out.println("Результат: false");

}

Функция

public static int same(double[]array){

int[] copy = new int[array.length];

int res = 0;

int countNum = 0;

int count = 0;

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

copy[i] = 0;

}

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

if (copy[i]==0){

for (int j = 0; j < array.length; j++) {

if (array[i]==array[j]){

copy[j] = 1;

count++;

}

}

}

if (countNum<count){

countNum = count;

res++;

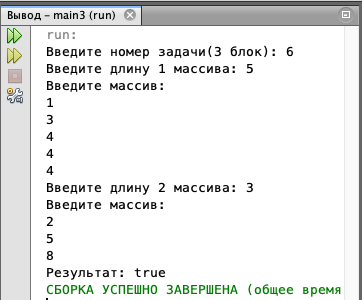
}

}

return res;

}

Результат:

****

**7 Задача:** Число Капрекара-это положительное целое число, которое после возведения в квадрат и разбиения на две лексикографические части равно сумме двух полученных новых чисел:

– Если количество цифр квадратного числа четное, то левая и правая части будут иметь одинаковую длину.a

– Если количество цифр квадратного числа нечетно, то правая часть будет самой длинной половиной, а левая-самой маленькой или равной нулю, если количество цифр равно 1.

– Учитывая положительное целое число n, реализуйте функцию, которая возвращает true, если это число Капрекара, и false, если это не так.

Решение:

Вызов

System.out.print("Введите число: ");

int k = sca.nextInt();

String res7 = isKaprekar(k);

System.out.println(res7);

Функция

public static String isKaprekar(int k){

String res7 = "[eq";

String Left;

String Right;

int k2 =0;

int l = 0;

int l1 = 0;

int r1 = 0;

String k1;

if ( k > 0 ){

k2 = k\*k;

l = String.valueOf(k2).length();

if (k2 < 10){

l1 = 0;

r1 = k2;

if (l1 + r1 == k){

res7 = "true";

}

else{

res7 = "false";

}

}else

if (l % 2 == 0 ){

k1 = Integer.toString(k2);

Left = k1.substring(0,l/2);

l1 = Integer.parseInt(Left.trim());

Right = k1.substring(l/2);

r1 = Integer.parseInt(Right.trim());

if (l1 + r1 == k){

res7 = "true";

}

else{

res7 = "false";

}

}else if(l % 2 != 0 ){

k1 = Integer.toString(k2);

Left = k1.substring(0,(l-1)/2);

l1 = Integer.parseInt(Left.trim());

Right = k1.substring((l-1)/2);

r1 = Integer.parseInt(Right.trim());

if (l1 + r1 == k){

res7 = "true";

}

else{

res7 = "false";

}

}

}else{

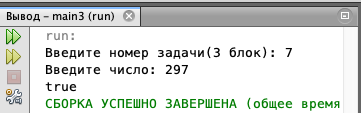
return " Введите другое число!";

}

return res7;

}

Результат:

****

**8 Задача:** Напишите функцию, которая возвращает самую длинную последовательность последовательных нулей в двоичной строке.

Решение:

Вызов

System.out.println("Введите двличную строку: ");

String s8 = sca.nextLine();

String res8 = longestZero(s8);

System.out.println(res8);

Функция

public static String longestZero(String s8){

int maxl = 1;

int curl = 1;

String res8 = "";

int one = 0;

for(int i = 0 ; i < s8.length()-1; i++){

if(s8.charAt(i) == '1'){

one+=1;

}

else

{

if (s8.charAt(i) == s8.charAt(i+1) ){

curl+=1;

}else{

if(curl > maxl){

maxl = curl;

}

curl = 1;

}

}

}

if (one == s8.length() -1 )

{

return res8;

}

if(curl > maxl){

maxl = curl;

}

for (int i = 0; i < maxl ; i++){

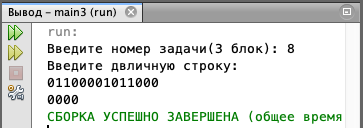
res8 += "0";

}

return res8;

}

Результат:

****

**9 Задача:** Если задано целое число, создайте функцию, которая возвращает следующее простое число. Если число простое, верните само число.

Решение:

Вызов

System.out.println("Введите число: ");

int a9 = sca.nextInt();

int res9 = nextPrime(a9);

System.out.println(res9);

Функция

public static int nextPrime(int a){

int res9 = 0;

int a1 = 0;

int a2 = a;

if (a == 1){

res9 = 1;

}else{

for(int i = 2; i <= a; i++){

if(a % i == 0){

a1++;

}

}

if(a1 == 1){

res9 = a;

}else{

do{

a1=0;

a2++;

for(int i = 1; i <= a2; i++){

if(a2 % i == 0){

res9 = a2;

a1++;

}

}

} while ( a1 != 2);

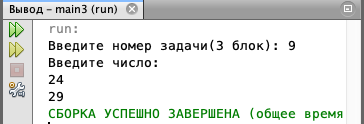
}

}

return res9;

}

Результат:



**10 Задача:** Учитывая три числа, x, y и z, определите, являются ли они ребрами прямоугольного треугольника.

Решение:

Вызов

System.out.println("Введите три числа:");

int n10 = sca.nextInt();

int n101 = sca.nextInt();

int n12 = sca.nextInt();

String res10 = rightTriangle(n10, n101, n12);

System.out.println(res10);

Функция

public static String rightTriangle(int n1,int n2,int n3){

int c=0,a=0,b=0;

String res10;

if ((n1 >= n2) && (n1 >= n3)){

c = n1;

a = n2;

b = n3;

}else if ((n2 >= n1) && (n2 >= n3)){

c = n2;

a = n1;

b = n3;

}else{

c = n3;

a = n2;

b = n1;

}

if (c\*c == (a\*a) + (b\*b)){

res10 = "true";

}else{

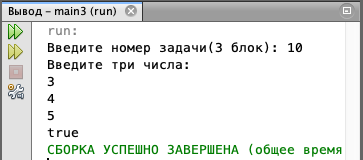
res10 = "false";

}

return res10;

}

Результат:



**Полный код программы:**

package main3;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

public class Main3 {

public static void main(String[] args) {

Scanner sca = new Scanner(System.in);

System.out.print("Введите номер задачи(3 блок): ");

int num = sca.nextInt();

sca.nextLine();

switch (num){

case 1://1 задача

System.out.println("Введите три значения квадратного уравнения");

int n11 = sca.nextInt();

int n21 = sca.nextInt();

int n31 = sca.nextInt();

int res1 = solutions(n11,n21,n31);

System.out.println("Число корней = " + res1);

break;

case 2://2 задача

System.out.print("Введите строку: ");

String st2 = sca.nextLine();

int res2 = findZip(st2);

System.out.println("Второе вхождение zip на " + res2 + " символе");

break;

case 3://3 задача

System.out.print("Введите число: ");

int n3 = sca.nextInt();

String res3 = checkPerfect(n3);

System.out.println("Результат: " + res3);

break;

case 4://4 задача

String st4 = new String();

System.out.println("Введите строку:");

st4 = sca.nextLine();

String res4 = flipEndChars(st4);

System.out.println(res4);

break;

case 5://5 задача

System.out.print("Введите строку: ");

String st5 = sca.nextLine();

String res5 = isValidHexCode(st5);

System.out.println("Результат: " + res5);

break;

case 6://6 задача

double [] x = new double[]{};

System.out.print("Введите длину 1 массива: ");

int s1 = sca.nextInt();

x = crarr(s1);

double [] y = new double[]{};

System.out.print("Введите длину 2 массива: ");

int s2 = sca.nextInt();

y = crarr(s2);

int x1 = same(x);

int y1 = same(y);

if (x1 == y1){

System.out.println("Результат: true");

}else{

System.out.println("Результат: false");

}

break;

case 7://7 задача

System.out.print("Введите число: ");

int k = sca.nextInt();

String res7 = isKaprekar(k);

System.out.println(res7);

break;

case 8://8 задача

System.out.println("Введите двличную строку: ");

String s8 = sca.nextLine();

String res8 = longestZero(s8);

System.out.println(res8);

break;

case 9://9 задача

System.out.println("Введите число: ");

int a9 = sca.nextInt();

int res9 = nextPrime(a9);

System.out.println(res9);

break;

case 10://10 задача

System.out.println("Введите три числа:");

int n10 = sca.nextInt();

int n101 = sca.nextInt();

int n12 = sca.nextInt();

String res10 = rightTriangle(n10, n101, n12);

System.out.println(res10);

break;

default:

System.out.println("Такой задачи в первом блоке нет!");

break;

}

}

//Функции

// Функция запонлнения массива

public static double[] crarr(int size){

Scanner sca1 = new Scanner(System.in);

double [] array = new double[size];

System.out.println("Введите массив: ");

for (int i = 0; i < size; i++)

{

array[i] = sca1.nextInt();

}

return array;

}

//задаяа 1

public static Integer solutions(int a, int b, int c){

int D = (b\*b) - 4 \* a \* c;

if(D > 0){

return 2;

}else if(D == 0){

return 1;

}else{

return 0;

}

}

//задача 2

public static int findZip(String st2){

// String SubStr = "Zip";

// int z = st2.indexOf(SubStr);

int z = st2.indexOf("zip", st2.indexOf("zip") + 1);

return z;

}

// задача 3

public static String checkPerfect(int n3)

{

int res3 = 0 ;

for (int i = 1; i< n3; i++){

if (n3 % i == 0 ){

res3 += i;

}

}

if (res3 == n3){

return "true";

} else {

return "false";

}

}

// задача 4

public static String flipEndChars (String st4){

String res4 = "";

if (st4.length() < 2){

res4 = "Incompatible.";

}

else if (st4.charAt(0) == st4.charAt(st4.length() - 1))

{

res4 = "Two's a pair.";

}

else

{

return st4.charAt(st4.length()-1) + st4.substring(1, st4.length()-1) + st4.charAt(0);

}

return res4;

}

//задача 5

public static String isValidHexCode(String st5){

int sum = 0;

String res5;

if (st5.charAt(0) == '#' && st5.length() == 7){

for (int i = 1; i <= 6 ; i++){

int a5 = ((int) st5.charAt(i));

if (((a5 >= 48) && (a5 <= 57)) ||((a5 >= 65) && (a5 <= 70)) ||((a5 >= 97) && (a5 <= 102)) ){

sum +=1;

}

}

}else{

res5 = "false";

}

if(sum == 6){

res5 = "true";

}else{

res5 = "false";

}

return res5;

}

//задача 6

public static int same(double[]array){

int[] copy = new int[array.length];

int res = 0;

int countNum = 0;

int count = 0;

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

copy[i] = 0;

}

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

if (copy[i]==0){

for (int j = 0; j < array.length; j++) {

if (array[i]==array[j]){

copy[j] = 1;

count++;

}

}

}

if (countNum<count){

countNum = count;

res++;

}

}

return res;

}

//задача 7

public static String isKaprekar(int k){

String res7 = "[eq";

String Left;

String Right;

int k2 =0;

int l = 0;

int l1 = 0;

int r1 = 0;

String k1;

if ( k > 0 ){

k2 = k\*k;

l = String.valueOf(k2).length();

if (k2 < 10){

l1 = 0;

r1 = k2;

if (l1 + r1 == k){

res7 = "true";

}

else{

res7 = "false";

}

}else

if (l % 2 == 0 ){

k1 = Integer.toString(k2);

Left = k1.substring(0,l/2);

l1 = Integer.parseInt(Left.trim());

Right = k1.substring(l/2);

r1 = Integer.parseInt(Right.trim());

if (l1 + r1 == k){

res7 = "true";

}

else{

res7 = "false";

}

}else if(l % 2 != 0 ){

k1 = Integer.toString(k2);

Left = k1.substring(0,(l-1)/2);

l1 = Integer.parseInt(Left.trim());

Right = k1.substring((l-1)/2);

r1 = Integer.parseInt(Right.trim());

if (l1 + r1 == k){

res7 = "true";

}

else{

res7 = "false";

}

}

}else{

return " Введите другое число!";

}

return res7;

}

//задача 8

public static String longestZero(String s8){

int maxl = 1;

int curl = 1;

String res8 = "";

int one = 0;

for(int i = 0 ; i < s8.length()-1; i++){

if(s8.charAt(i) == '1'){

one+=1;

}

else

{

if (s8.charAt(i) == s8.charAt(i+1) ){

curl+=1;

}else{

if(curl > maxl){

maxl = curl;

}

curl = 1;

}

}

}

if (one == s8.length() -1 )

{

return res8;

}

if(curl > maxl){

maxl = curl;

}

for (int i = 0; i < maxl ; i++){

res8 += "0";

}

return res8;

}

//задача 9

public static int nextPrime(int a){

int res9 = 0;

int a1 = 0;

int a2 = a;

if (a == 1){

res9 = 1;

}else{

for(int i = 2; i <= a; i++){

if(a % i == 0){

a1++;

}

}

if(a1 == 1){

res9 = a;

}else{

do{

a1=0;

a2++;

for(int i = 1; i <= a2; i++){

if(a2 % i == 0){

res9 = a2;

a1++;

}

}

} while ( a1 != 2);

}

}

return res9;

}

//задача 10

public static String rightTriangle(int n1,int n2,int n3){

int c=0,a=0,b=0;

String res10;

if ((n1 >= n2) && (n1 >= n3)){

c = n1;

a = n2;

b = n3;

}else if ((n2 >= n1) && (n2 >= n3)){

c = n2;

a = n1;

b = n3;

}else{

c = n3;

a = n2;

b = n1;

}

if (c\*c == (a\*a) + (b\*b)){

res10 = "true";

}else{

res10 = "false";

}

return res10;

}

}