**«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПУТЕЙ СООБЩЕНИЯ (МИИТ) императора Николая ll»**

**ИНСТИТУТ УПРАВЛЕНИЯ ИНФОРМАЦИОННЫМИ ТЕХНОЛОГИЯМИ**

**Кафедра «Вычислительные системы и сети»**

**ОТЧЕТ**

**Индивидуальное задание студента группы** УИБ-211

**Евстропова Николая Геннадьевича**

По дисциплине “Технологии программирования ”

Приняли:

Москва ­– 2016

***Программа 1:***

Задание : вычислить данный ряд Z=(b^(x+0.5))/(sin(x-15)+ (b^(2x+1))/(cos(x-20)+(b^(4x+1.5))/(sin(x-25)…

Блок вывода данных

Блок обработки данных

Блок ввода данных

Текст программы:

package ряд;

import java.util.InputMismatchException;

import java.util.Scanner;

public class ряд1 {

public static void main (String[] args) {

double b = 0.1,x = 0.01,E = 0.000000001,n = 0.5,Z,znamenatel;

int kalibr = 9;

@SuppressWarnings("resource")

Scanner input = new Scanner(System.in);

System.out.println("введите b ");

b = input.nextDouble();

System.out.println("введите x ");

x = input.nextDouble();

System.out.println("введите kalibr ");

kalib = input.nextInt();

znamenatel = Math.sin(Math.toRadians(x-(n\*10+10)));//подсчет знаменателя ряда

Z=Math.pow(b,(x+n))/znamenatel;//подсчет элементов ряда

System.out.println("первый элемент"+Z);

int i = 0;

while(Math.abs(Z) > E&&i <= kalibr-1){//цикличная проверка условия вычислений ряда

n = n+0.5;

i++;

znamenatel=smena(znamenatel,x,n);

Z = Z+Math.pow(b,x\*2\*i+n)/znamenatel;

System.out.println("промежуточный результат "+Z+" "+i);

}

if(i==kalibr){//вывод сообщения на выход из вычислений по задаваемому нами числу

System.out.println("ряд не сходится при x= "+x+" и b= "+b+"точность по эпсилон= "+E+"не достигнуто расчет закончен на элементе номер "+kalibr+"заданный нами");

}

else//результат вычислений

System.out.println("ряд сходится и имеет результат = "+Z+" при эпсилон "+E+"и при заданном номере элемента конца"+kalibr);

}

public static double smena(double b,double x,double n)//чередуем синус с косинусом

{

if(b==Math.sin(Math.toRadians(x-((n-0.5)\*10+10))))

{

return Math.cos(Math.toRadians(x-(n\*10+10)));

}

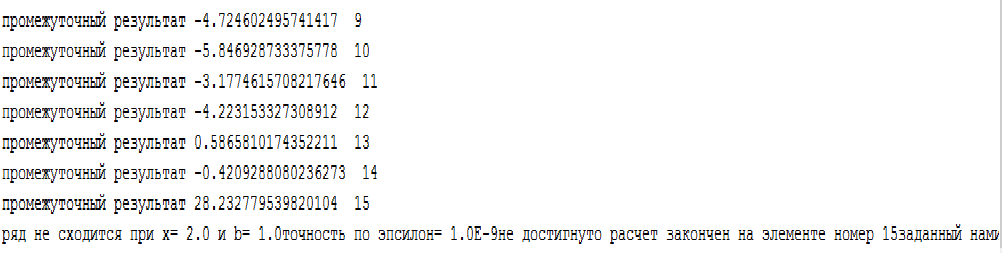
else

return Math.sin(Math.toRadians(x-(n\*10+10)));

}

}

Результат работы программы



**Ряд сходится не достигнув максимального количество элементов заданного нами**



***Программа 2:***

Задание : сформировать таблицу данных имеющую структуру

|название города|количество районов|название района|площадь района|насаление района|

Блок вывода данных

Блок обработки данных

Блок ввода данных

***Текст программы***:

package tabl;

import java.io.File;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.util.Scanner;

public class Citi {

private static int kolGor = 1;

Citi(String[] args) {

Scanner SCN = new Scanner(System.in);

System.out.println("введите количество городов");

kolGor = SCN.nextInt();

String []gorod = new String[kolGor];

double plosad[][] = new double[kolGor][];

String NazR[][] = new String[kolGor][];

double naselenie[][] = new double[kolGor][];

int obshKol=0;

double obshPLO = 0;

int j = 0;

double minR = Double.MAX\_VALUE;

for(;j<kolGor;j++){

System.out.println("введите название города");

gorod[j] = SCN.next();

try(FileWriter writer = new FileWriter(gorod[j]+".txt", false))

{

// запись в файл

System.out.println("введите количество районов района");

String KR = SCN.next();

writer.write(KR);

writer.append('\n');

for(int k = 0 ; k < Integer.parseInt(KR) ; k ++){

System.out.println("введите название района");

String NR = SCN.next();

System.out.println("введите плошадь района");

String PR = SCN.next();

System.out.println("введите население района");

String NASR = SCN.next();

writer.write(NR);

writer.append(' ');

writer.write(PR);

writer.append(' ');

writer.write(NASR);

writer.append('\n');

writer.flush();

}

}

catch(IOException ex){

System.out.println(ex.getMessage());

}

}

int i = 0;

int index = 0;

System.out.println("данные записаные в таблицу ");

System.out.println("название города | количество районов |имя района |площадь района |кол населения|");

while(i<kolGor){

File f=new File(gorod[i]+".txt");

try(FileReader reader = new FileReader(f))

{

char[] buffer = new char[(int)f.length()];

// считаем файл полностью

reader.read(buffer);

String g=new String(buffer);

int r=kolR(g);

obshKol=+r;

plosad[i] = new double[r];

naselenie[i] = new double[r];

NazR[i] = new String[r];

String inf[] = new String[r];

//делим данные из файла на строки разделяющиеся /n

inf = g.split("\n");

System.out.printf("%s| %15d| ",gorod[i],r);

for(int k = 0 ; k < r ; k++){

//заносим данные из файла в масивы

String naz = inf[k+1].substring(0, indexOf(inf[k+1],' '));

double PLO = Double.parseDouble(inf[k+1].substring(indexOf(inf[k+1],' '), lastIndexOf(inf[k+1],' ')));

double NAS = Double.parseDouble(inf[k+1].substring(lastIndexOf(inf[k+1],' '), inf[k+1].length()));

plosad[i][k]=PLO;

obshPLO=+PLO;

naselenie[i][k]=NAS;

NazR[i][k]=naz;

// System.out.println(" "+" | "+" | "+NazR[i][k]+" | "+plosad[i][k]+" | "+naselenie[i][k]+"|");

System.out.printf("%26s| %15f| %15f| ",NazR[i][k],plosad[i][k],naselenie[i][k]);

System.out.println();

}

System.out.println("|----------------------------------------------------------------------------------------------------------------|");

//находим минимальное количество районов и индекс города с мин кол районов

if(minR>r){

minR=r;

index = i;

}

i++;

}

catch(IOException ex){

System.out.println(ex.getMessage());

}

}

System.out.println("город с минимальным количеством районов <<"+gorod[index]+">> с количеством районов "+(int)minR);

System.out.println("районы с площадью ниже средней среднее равно "+(obshPLO/obshKol));

for(i = 0 ; i < minR ; i++){

//нахождение районов с площадью меньше средней

if((obshPLO/obshKol)>plosad[index][i]){

System.out.println("название района <<"+NazR[index][i]+">> его площадь <<"+plosad[index][i]+">>") ;

}

else if(i==minR){

System.out.println(" этом городе нет районов с плошадью меньше средней");

}

}

}

//функция по нахождению первого входа символа ch в строку string

private static int indexOf(String string,char ch) {

int i = 0;

while(true){

if(ch==string.charAt(i))

return i;

i++;

}

}

//функция по нахождению отличного от первого входа символа ch в строку string

private static int lastIndexOf(String string,char ch) {

int k = indexOf(string,ch)+1;

while(true){

if(ch==string.charAt(k))

return k;

k++;

}

}

//функция по переводу стринговой переменной в интовскую : обозначабщей количество районов города

public static int kolR(String k){

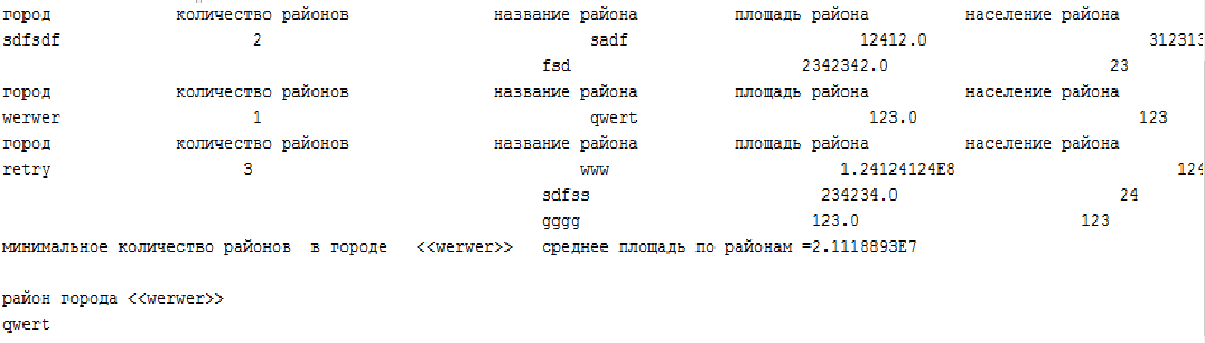
String kol = k.substring(0,1);

return Integer.parseInt(kol);

}

}

Результат работы



***Программа 3:***

Задание :составить программу по условию задания №2.Отличие состоит в том, что каждая строка исходной таблицы должна быть представлена виде объекта.

Блок ввода данных

Блок обработки данных

Блок вывода данных

Текст программы

Класс как структура

package tabl;

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Paths;

import java.util.Scanner;

public class CitiStructure {

//исходные данные класа выступающего структурой

private int KolRaion = 1;

private String gorod;

private String[] Raion ;

private double[] RaionRO ;

private long[] RaionNAS ;

public void vivod(){//функция вывода

System.out.println("город количество районов название района площадь района население района");

System.out.print(gorod+" "+KolRaion);

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

if(i>0){

System.out.println(" "+Raion[i]+" "+RaionRO[i]+" "+RaionNAS[i]);

}

else

System.out.println(" "+Raion[i]+" "+RaionRO[i]+" "+RaionNAS[i]);

}

}

public void vvod(){//функция ввода

Scanner scn = new Scanner(System.in);

try{// отработка исключений при вводе названия города и количества районов

System.out.println("введите название города");

gorod = scn.nextLine();

System.out.println("введите количество районов");

KolRaion = scn.nextInt();

}catch(Exception e){

System.out.println("ошибка ");

}

//инициализация масивов

Raion = new String[KolRaion];

RaionRO = new double[KolRaion];

RaionNAS = new long[KolRaion];

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

try{//отработка исключающих случаев при вводе названия района

System.out.println("введите название района");

Raion[i] = scn.next();

}catch(Exception e){

System.out.println("введеное название района привысила лимит next ");

}

try{//отработка исключающих случаев при вводе площади района

System.out.println("введите площадь района");

RaionRO[i] = scn.nextDouble();

}catch(Exception e){

System.out.println("введеная площадь района привысила лимит double ");

RaionRO[i] = Double.MAX\_VALUE;//при исключении присваиваем максимальное значение double

}

try{//отработка исключающих случаев при вводе населения района

System.out.println("введите население района");

RaionNAS[i] = scn.nextLong();

}catch(Exception e){

System.out.println("введеная население района привысила лимит long ");

RaionNAS[i] = Long.MAX\_VALUE;//при искллючение присваиваем максимальное значение long

}

}

}

//геттеры и сеттеры

public int getKolRaion(){

return KolRaion;

}

public String getGorod() {

return gorod;

}

public String[] getRaion() {

return Raion;

}

public double[] getRaionRO() {

return RaionRO;

}

public long[] getRaionNAS() {

return RaionNAS;

}

}

Класс с функциями обработки данных

**package** tabl;

**import** java.util.Scanner;

**public** **class** Citi {

//исходные данные в основном классе с default вариантом значения

**private** **static** **int** kolGor=2,ind=0;

**public** **static** **void** main(String[] args) {

FilesApp e = **new** FilesApp();

e.fileTest("1");

Scanner scn = **new** Scanner(System.in);

System.out.println("введите количество городов");

kolGor = scn.nextInt();//вводим количество городов

CitiStructure []a = **new** CitiStructure[kolGor];//объявление количества объектов класса сруктуры

**for**(**int** i = 0 ; i < kolGor ; i++)

a[i]=**new** CitiStructure();//инициализация объектов класса структуры

**for**(**int** i = 0 ; i < kolGor ; i++){//вводим информацию на все города

System.out.println();

System.out.println("город № "+(i+1));

System.out.println();

a[i].vvod();

}

**for**(**int** i = 0 ; i < kolGor ; i++)//выводим информацию на все города

a[i].vivod();

System.out.println("минимальное количество районов в городе <<"+MinR(a)+">> среднее площадь по районам ="+SredRRO(a));

**boolean** flag = **true**;//объявляем флаг на случай если нет районов с площадью меньше средней

**for**(**int** i = 0 ; i < a[ind].getKolRaion() ; i++){

System.out.println();

**if**(SredRRO(a)>a[ind].getRaionRO()[i]){

flag = **false**;

System.out.println("район города <<"+MinR(a)+">>");

System.out.println(a[ind].getRaion()[i]);//вывод района с площадью меньше средней

}

**if**(flag)//вывод если в городе с минимальным количеством районов нет районов с площадью меньше средней площади по всем районам

System.out.println("этот город не имеет районов с площадью меньше средней ");

}

}

**public** **static** String MinR(CitiStructure [] f){//функция пол нахождению города с минимальным количеством районов

**int** MinR=f[0].getKolRaion();

**for**(**int** i = 1 ; i < kolGor ; i++)

**if**(MinR>f[i].getKolRaion()){

MinR=f[i].getKolRaion();

ind =i;

}

**return** f[ind].getGorod();

}

**public** **static** **double** SredRRO(CitiStructure [] f){//функция нахождения среднего значения площади всех районов

**double** srednee = 0,kolR = 0;

**for**(**int** i = 0 ; i < kolGor ; i++){

kolR+=f[i].getKolRaion();

**for**(**int** j = 0 ; j < f[i].getKolRaion() ; j++){

srednee+=f[i].getRaionRO()[j];

}

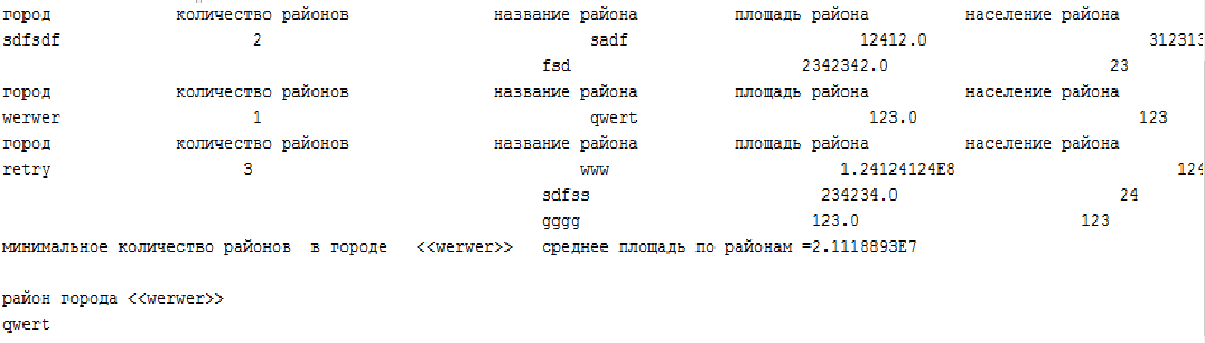
}

**return** srednee/kolR;

}

}

Результат работы



***Программа 4:***

Задание : проанализировать структуру StringBuffer и составлена программа тестирования всех констант, включенных в класс(если они имеются) и методов , принадлежащих этому классу

Текст программы

**package** StringBuffer;

Блок вывода результатов иследования

**public** **class** str {

**public** **static** **void** main(String[] args) {

StringBuffer strB = **new** StringBuffer();

StringBuffer strA = **new** StringBuffer();

StringBuffer strAB ;

StringBuffer strk[] ;

Double g ;

g=Double.***MIN\_NORMAL***;

System.***out***.println(g);

String str1 = "retry kon 2016";

System.***out***.println("String str1 = retry kon 2016;");

strAB= **new** StringBuffer(str1);

System.***out***.println("strAB= new StringBuffer(str); "+strAB);

strk = **new** StringBuffer[2];

**int** n = strk.length;

System.***out***.println("int n = strk.length = "+n);

**for**(**int** i = 0 ; i < 2 ;i ++){

System.***out***.println("strk[i] = "+strk[i]);

}

strk[0]=**new** StringBuffer(str1+" strk[0]");

strk[1]=**new** StringBuffer(str1+" strk[1]");

System.***out***.println("strk = new StringBuffer[2];\n strk[0]=new StringBuffer(str1+strk[0]);\nstrk[1]=new StringBuffer(str1+strk[1]);");

**for**(**int** i1 = 0 ; i1 < 2 ;i1 ++){

System.***out***.println("strk[i] = "+strk[i1]);

}

System.***out***.println("StringBuffer.class.getEnumConstants() "+StringBuffer.**class**.getEnumConstants());

System.***out***.println("StringBuffer.class.getName() "+StringBuffer.**class**.getName());

System.***out***.println("StringBuffer.class.getCanonicalName() "+StringBuffer.**class**.getCanonicalName());

System.***out***.println("StringBuffer.class.getModifiers() "+StringBuffer.**class**.getModifiers());

System.***out***.println("StringBuffer.class.getSimpleName() "+StringBuffer.**class**.getSimpleName());

System.***out***.println("StringBuffer.class.hashCode() "+StringBuffer.**class**.hashCode());

System.***out***.println("StringBuffer.class.toString() "+StringBuffer.**class**.toString());

System.***out***.println("StringBuffer.class.isArray() "+StringBuffer.**class**.isArray());

System.***out***.println("StringBuffer.class.getDeclaredMethods() "+StringBuffer.**class**.getDeclaredMethods());

strB.append(**true**);

System.***out***.println("strB.append(true);Boolean: "+strB);

strB.append(89);

System.***out***.println("strB.append(89);int: "+strB);

strB.append(" double ");

System.***out***.println("strB.append( double );Sring: "+strB);

strB.append(Double.***MAX\_VALUE***);

System.***out***.println("strB.append(Double.MAX\_VALUE);double: "+strB);

strB.append('L');

System.***out***.println("strB.append('L');char: "+strB);

strB.append(Long.***MAX\_VALUE***);

System.***out***.println("strB.append(Long.MAX\_VALUE);long: "+strB);

System.***out***.println("strB.charAt(7) "+strB.charAt(7));

**char** []str = **new** **char**[4];

str[0]='r';

str[1]='e';

str[2]='t';

str[3]='y';

strAB.append(str);

strAB.trimToSize();

System.***out***.println("char []str = new char[4];\nstr[0]='r';\nstr[1]='e';\nstr[2]='t';\nstr[3]='y'; ");

System.***out***.println("strAB.append(str);char[]: "+strAB);

System.***out***.println("strAB.reverse() "+strAB.reverse());

String S=strB.toString();

strA.append(S);

System.***out***.println("вывод String "+S);

**int** c = strB.capacity();

**int** cP =strB.codePointBefore(7);

System.***out***.println("strB.capacity: "+c+" strB.codePointBefore(7): "+cP);

strB.deleteCharAt(0);

System.***out***.println("strB.deleteCharAt(0) "+strB);

strB.delete(10, 15);

System.***out***.println("strB.delete(10, 15) "+strB);

strB.codePointCount(5, 10);

System.***out***.println("strB.codePointCount(5, 10) "+strB);

System.***out***.println("String S=strB.toString();\nstrA.append(S);\nstrA.equals(S) "+strA.toString().equals(S));

strA=strB;

System.***out***.println("strA=strB\nstrA.equals(strB) "+strA.equals(strB));

strB.ensureCapacity(2);

System.***out***.println("strB.ensureCapacity(2) "+strB);

System.***out***.println("strB.codePointAt(3) "+strB.codePointAt(3));

strB.appendCodePoint(56);

System.***out***.println("strB.appendCodePoint(56) "+strB);

System.***out***.println("strB.hashCode() "+strB.hashCode());

System.***out***.println("strB.insert(cP, false) "+strB.insert(8, **false**));

System.***out***.println("replace(1, 4, kol) "+strB.replace(1, 4, "kol"));

System.***out***.println("strB.subSequence(6, 9) "+strB.subSequence(6, 9));

**int** k = strB.offsetByCodePoints(3, 9);

System.***out***.println("offsetByCodePoints(3, 9) "+k);

System.***out***.println("strB.substring(strB.indexOf(\0),strB.lastIndexOf(7)) "+strB.substring(strB.indexOf(""),strB.lastIndexOf("7")));

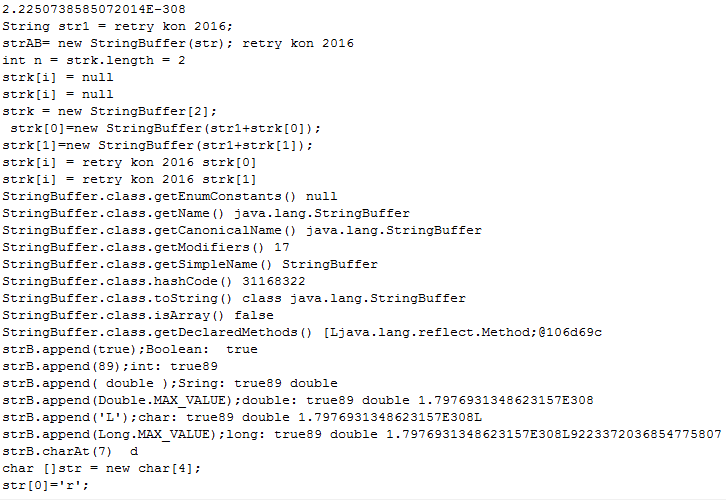
strB.setLength(15);

System.***out***.println("strB.setLength(15) "+strB);

}

}

***Результат работы***



***Программа 5 :***

Задание: составить программу , которая выполняет суммирование ряда , заданного в первом задание. Отличие состоит в том, что если было задано х по абсолютному значению больше Integer.MAX\_VALUE , то создается исключающая ситуация , при обработке которой новое значение х получается путем вычитания из абсолютного значения х шага l до получения Integer.MAX\_VALUE

Блок вывода данных

Блок обработки данных

Блок обработки исключающего

Блок ввода данных

**Текст программы**

**package** обработкаИсключающего;

**import** java.util.InputMismatchException;

**import** java.util.Scanner;

**public** **class** ряд1 {

**public** **static** **void** main (String[] args) {

**double** b = 0.1,x = 0.01,E = 0.000000001,n = 0.5,Z,znamenatel;

**int** kalib,kalibr = 9;

@SuppressWarnings("resource")

Scanner input = **new** Scanner(System.***in***);

**try**//вводим значения b,x;kalibr-задаваемое количество элементов ряда

{

System.***out***.println("введите b ");

b = input.nextDouble();

}**catch**(InputMismatchException e){

System.***out***.println("ошибка ");

**if**(b>Double.***MAX\_VALUE***){

System.***out***.println("введите шаг с которым будем уменьшаь значание до удовл условия b<Double.MAX\_VALUE");

**int** interv = input.nextInt();

**if**(interv>Integer.***MAX\_VALUE***){

interv = Integer.***MAX\_VALUE***;

}

**while**(**true**){

b=b-interv;

**if**(b<Double.***MAX\_VALUE***){

**break**;

}

}

}

};

**try**

{

System.***out***.println("введите x ");

x = input.nextDouble();

}**catch**(InputMismatchException e){

System.***out***.println("ошибка ");

**if**(b>Double.***MAX\_VALUE***){

String g =e.toString();

**double** krag = Double.*parseDouble*(g.substring(g.indexOf('"')+1, g.lastIndexOf('"')));

System.***out***.println("ошибка ");

System.***out***.println("введите шаг с которым будем уменьшаь значание до удовл условия b<Double.MAX\_VALUE");

Scanner input1 = **new** Scanner(System.***in***);

**int** interv = input1.nextInt();

**if**(interv>Integer.***MAX\_VALUE***){

interv = Integer.***MAX\_VALUE***;

}

**while**(**true**){

x=x-interv;

**if**(x<Double.***MAX\_VALUE***){

**break**;

}

}

}

};

**try**

{

System.***out***.println("введите kalibr ");

kalib = input.nextInt();

kalibr = kalib;

}**catch**(InputMismatchException e){

String g =e.toString();

**double** krag = Double.*parseDouble*(g.substring(g.indexOf('"')+1, g.lastIndexOf('"')));

System.***out***.println("ошибка ");

**if**(krag>(**double**)Integer.***MAX\_VALUE***){

Scanner input1 = **new** Scanner(System.***in***);

System.***out***.println("введите шаг с которым будем уменьшаь значание до удовл условия b<Double.MAX\_VALUE");

**double** interv = input1.nextDouble();

**while**(**true**){

krag=krag-interv;

**if**(krag<(**double**)Integer.***MAX\_VALUE***){

kalibr = (**int**)krag;

**break**;

}

}

}

};

znamenatel = Math.*sin*(Math.*toRadians*(x-(n\*10+10)));//подсчет знаменателя ряда

Z=Math.*pow*(b,(x+n))/znamenatel;//подсчет элементов ряда

System.***out***.println("первый элемент"+Z);

**int** i = 0;

**while**(Math.*abs*(Z) > E&&i <= kalibr-1){//цикличная проверка условия вычислений ряда

n = n+0.5;

i++;

znamenatel=*smena*(znamenatel,x,n);

Z = Z+Math.*pow*(b,x\*2\*i+n)/znamenatel;

System.***out***.println("промежуточный результат "+Z+" "+i);

}

**if**(i==kalibr){//вывод сообщения на выход из вычислений по задаваемому нами числу

System.***out***.println("ряд не сходится при x= "+x+" и b= "+b+"точность по эпсилон= "+E+"не достигнуто расчет закончен на элементе номер "+kalibr+"заданный нами");

}

**else**//результат вычислений

System.***out***.println("ряд сходится и имеет результат = "+Z+" при эпсилон "+E+"и при заданном номере элемента конца"+kalibr);

}

**public** **static** **double** smena(**double** b,**double** x,**double** n)//чередуем синус с косинусом

{

**if**(b==Math.*sin*(Math.*toRadians*(x-((n-0.5)\*10+10))))

{

**return** Math.*cos*(Math.*toRadians*(x-(n\*10+10)));

}

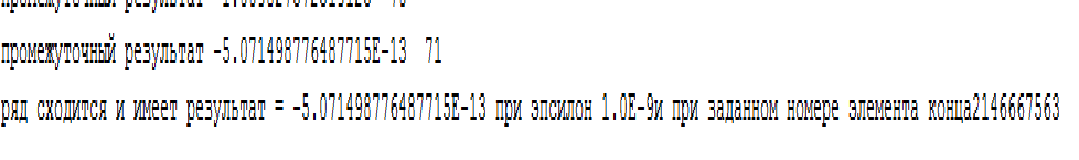
**else**

**return** Math.*sin*(Math.*toRadians*(x-(n\*10+10)));

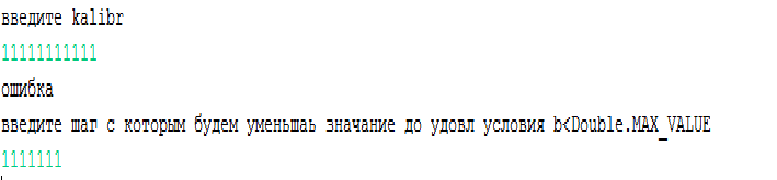
}

}

Результат работы программы



Результат работы обработки исключающего



***Программа 6:***

Задание пользуясь заданием программы 2 используя

механизм наследования

Блок Ввода данных

Блок обработки данных

Блок вывода результата

Текст программы

Класс наследник

**package** Nasled;

**import** java.util.Scanner;

**class** a{

a(){

CitiStructure s = **new** CitiStructure();

Citi c = **new** Citi();

s.KolRaion=3;

System.***out***.println(s);

c.KolRaion=0;

System.***out***.println(c);

}

}

**public** **class** Citi **extends** CitiStructure{

//исходные данные в основном классе с default вариантом значения

**private** **static** **int** *kolGor*=2,*ind*=0;

**public** **static** **void** main(String[] args) {

a r = **new** a();

Scanner scn = **new** Scanner(System.***in***);

System.***out***.println("введите количество городов");

*kolGor* = scn.nextInt();//вводим количество городов

Citi []a = **new** Citi[*kolGor*];//объявление количества объектов класса сруктуры

**for**(**int** i = 0 ; i < *kolGor* ; i++)

a[i]=**new** Citi();//инициализация объектов класса структуры

**for**(**int** i = 0 ; i < *kolGor* ; i++){//вводим информацию на все города

System.***out***.println();

System.***out***.println("город № "+(i+1));

System.***out***.println();

a[i].vvod();

}

**for**(**int** i = 0 ; i < *kolGor* ; i++)//выводим информацию на все города

a[i].vivod();

System.***out***.println("минимальное количество районов в городе <<"+*MinR*(a)+">> среднее площадь по районам ="+*SredRRO*(a));

**boolean** flag = **true**;//объявляем флаг на случай если нет районов с площадью меньше средней

**for**(**int** i = 0 ; i < a[*ind*].KolRaion ; i++){

System.***out***.println();

**if**(*SredRRO*(a)>a[*ind*].RaionRO[i]){

flag = **false**;

System.***out***.println("район города <<"+*MinR*(a)+">>");

System.***out***.println(a[*ind*].Raion[i]);//вывод района с площадью меньше средней

}

**if**(flag)//вывод если в городе с минимальным количеством районов нет районов с площадью меньше средней площади по всем районам

System.***out***.println("этот город не имеет районов с площадью меньше средней ");

}

}

**private** **static** String MinR(CitiStructure [] f){//функция пол нахождению города с минимальным количеством районов

**int** MinR=f[0].KolRaion;

**for**(**int** i = 1 ; i < *kolGor* ; i++)

**if**(MinR>f[i].KolRaion){

MinR=f[i].KolRaion;

*ind* =i;

}

**return** f[*ind*].gorod;

}

**private** **void** vivod(){//функция вывода

System.***out***.println("город количество районов название района площадь района население района");

System.***out***.print(gorod+" "+KolRaion);

**for**(**int** i = 0 ; i < KolRaion ; i++){

**if**(i>0){

System.***out***.println(" "+Raion[i]+" "+RaionRO[i]+" "+RaionNAS[i]);

}

**else**

System.***out***.println(" "+Raion[i]+" "+RaionRO[i]+" "+RaionNAS[i]);

}

}

**private** **void** vvod(){//функция ввода

Scanner scn = **new** Scanner(System.***in***);

**try**{// отработка исключений при вводе названия города и количества районов

System.***out***.println("введите название города");

gorod = scn.nextLine();

System.***out***.println("введите количество районов");

KolRaion = scn.nextInt();

}**catch**(Exception e){

System.***out***.println("ошибка ");

}

//инициализация масивов

Raion = **new** String[KolRaion];

RaionRO = **new** **double**[KolRaion];

RaionNAS = **new** **long**[KolRaion];

**for**(**int** i = 0 ; i < KolRaion ; i++){

**try**{//отработка исключающих случаев при вводе названия района

System.***out***.println("введите название района");

Raion[i] = scn.next();

}**catch**(Exception e){

System.***out***.println("введеное название района привысила лимит next ");

}

**try**{//отработка исключающих случаев при вводе площади района

System.***out***.println("введите площадь района");

RaionRO[i] = scn.nextDouble();

}**catch**(Exception e){

System.***out***.println("введеная площадь района привысила лимит double ");

RaionRO[i] = Double.***MAX\_VALUE***;//при исключении присваиваем максимальное значение double

}

**try**{//отработка исключающих случаев при вводе населения района

System.***out***.println("введите население района");

RaionNAS[i] = scn.nextLong();

}**catch**(Exception e){

System.***out***.println("введеная население района привысила лимит long ");

RaionNAS[i] = Long.***MAX\_VALUE***;//при искллючение присваиваем максимальное значение long

}

}

}

**protected** **static** **double** SredRRO(CitiStructure [] f){//функция нахождения среднего значения площади всех районов

**double** srednee = 0,kolR = 0;

**for**(**int** i = 0 ; i < *kolGor* ; i++){

kolR+=f[i].KolRaion;

**for**(**int** j = 0 ; j < f[i].KolRaion ; j++){

srednee+=f[i].RaionRO[j];

}

}

**return** srednee/kolR;

}

}

Супер класс родитель

**package** Nasled;

**import** java.io.IOException;

**import** java.nio.file.Files;

**import** java.nio.file.Paths;

**import** java.util.Scanner;

**class** CitiStructure {

//исходные данные клаcса выступающего структурой

**int** KolRaion = 1;

**protected** String gorod;

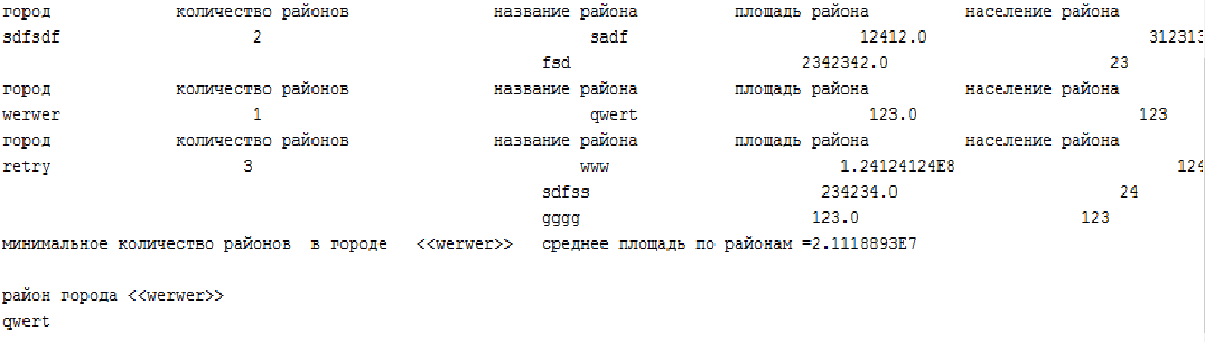
**protected** String[] Raion ;

**protected** **double**[] RaionRO ;

**protected** **long**[] RaionNAS ;

}

Результат работы программы



Попытка работы с данными супер класса сторонним классом



***Программ 7:***

Задание: составить программу создающую таблицу , которая содержит строки из двух полей: поле ключа и информационное поле. Программа выполняет запись данных в две таблицы , поиск строки с заданным ключем

##### | название город|количество районов|название района|площадь района|население района|

Используя для хранения таблицы высокоуровневый контейнер Hashtable<key,value>

package hashtable;

Блок вывода данных

Блок обработки данных

Блок ввода данных

Блок вывода данных из Hashtabl

import java.util.Collection;

import java.util.Enumeration;

import java.util.Hashtable;

import java.util.Iterator;

import java.util.Map;

import java.util.Map.Entry;

import java.util.Scanner;

import java.util.Set;

import javax.lang.model.element.Element;

public class hashtabl {

static int kolg;

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

Hashtable<String,String> citi = new Hashtable<String,String>();

int kolr;

String ng = null,nr;

double ro,nas;

System.out.println("введите количество городов");

kolg = s.nextInt();

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

System.out.println("введите название города");

ng = s.next();

System.out.println("введите количество районов");

kolr = s.nextInt();

String str = "";

for(int j = 0 ; j < kolr ; j++){

System.out.println("введите название района");

nr = s.next();

System.out.println("введите площадь района");

ro = s.nextDouble();

System.out.println("введите население района");

nas = s.nextDouble();

str+=nr+"|"+ro+"|"+nas+"\n";

}

citi.put(ng,kolr+"|"+str+":");

}

System.out.println("название города количество районов название района площадь района население района");

Enumeration g =citi.keys();

int kol = 0;

while(g.hasMoreElements()){

String gorod = g.nextElement().toString();

System.out.println(gorod+"="+citi.get(gorod));

kol++;

}

obrabotka(kol,citi);

}

static void obrabotka( int kol , Hashtable citi){

Enumeration<String> go =citi.keys();

String ng[] = new String[kol];

int kolr[] = new int[kol];

String nr[][] = new String[kol][];

double obshS , obnas , obro;

double ro[][] = new double[kol][];

double nas[][] = new double[kol][];

int koll=0;

while(go.hasMoreElements()){

String gorod = go.nextElement().toString();

ng[koll]=gorod;

String val =citi.get(gorod).toString();

char[] c=val.toCharArray();

int i = 1;

int d=0,l=Integer.parseInt(val.substring(0, 1));

kolr[koll]=l;

nr[koll] = new String[l];

ro[koll] = new double[l];

nas[koll] = new double[l];

int beg = 0 , end = 0;

while(d<l){

if(d==0)

beg = indexof(c,beg,false);

end = indexof(c,beg+1,false);

nr[koll][d]=val.substring(beg+1,end);

beg = end;

end = indexof(c,beg+1,false);

ro[koll][d]=Double.parseDouble(val.substring(beg+1,end));

beg = end;

end = indexof(c,beg+1,true);

nas[koll][d]=Double.parseDouble(val.substring(beg+1,end));

beg = end+1;

d++;

}

koll++;

}

int min =minR(kolr);

outRayonInCiti(min, sred(ro, kolr), sredR(nas[min], ro[min] , kolr[min]), ng[min] , kolr[min],nr[min]);

}

static int indexof(char[] c,int index,boolean flag){

if(flag){

while(c[index]!='\n'&&c[index+1]!=':')

index++;

return index;

}

else{

while(c[index]!='|')

index++;

return index;

}

}

static double[]sredR( double [] nas , double [] ro , int k){

double [] s = new double[k];

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

s[i]=ro[i]/nas[i];

}

return s;

}

static void outRayonInCiti(int ind , double Oro , double [] sr , String ng , int kol , String []nr){

boolean flag = true;

System.out.println("город с минимальным количеством районов "+ng);

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

if(Oro>sr[i]){

flag = false;

System.out.println("район с площадью на население меньше общей площади на население "+nr[i]);

}

}

if(flag)

System.out.println("нет таких районов ");

}

static int minR(int []kol){

int k = kol.length , i = 1,ind = 0;

while(i<k-1){

if(kol[ind]>kol[i+1]){

ind = i+1;

}

i++;

}

return ind;

}

static double sred(double [][]sred , int []kol){

int k = 0 ,l = 0 , koll = kol.length , kk = 0;

double roob = 0;

double roloc[] = new double[koll];

while(l<koll){

kk += kol[l];

while(k<kol[l]){

roloc[l] +=sred[l][k];

k++;

}

roob += roloc[l];

l++;

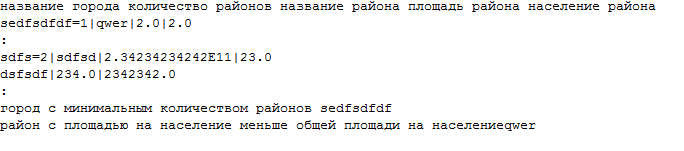
}

return roob/kk;

}

}

**Результат работы программы:**



***Программа 8:***

Задание: написать программу с пользовательским графическим интерфейсом которая будет шифровать текст и расшифровывать по методу Вежинера.

***Текст программы***

Choice ---переключение шифрование или расшифрование;

FlowLayout----менеджер расположения на окне;

JButton------кнопка;

JFrame------наша форма;

JLabel-------- отображение текста;

JPasswordField-----------поле ввода текста с отображением вместа текста одного символа;

JTextField-------------поле ввода текста

Блок вывода результата в созданном окне

Блок обработки данных

Блок создания окна добавления в него элементов

package frame;

import java.awt.Choice;

import java.awt.FlowLayout;

import java.awt.LayoutManager;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPasswordField;

import javax.swing.JTextField;

public class qq extends JFrame {

static JButton b = new JButton("actian");

static JPasswordField ps = new JPasswordField(8);

static JTextField fl = new JTextField(8);

static JTextField res = new JTextField("результат",8);

static Choice ch = new Choice();

static JLabel l1 , l2 , result;

public static void main(String[] args) {

javax.swing.SwingUtilities.invokeLater(new Runnable() {

public void run() {

l1 = new JLabel("текст");

l2 = new JLabel("ключ");

result = new JLabel(" ");

JFrame.setDefaultLookAndFeelDecorated(true);

JFrame f = new JFrame("шифратор");

f.setLayout(new FlowLayout());

ch.addItem("shifr");

ch.addItem("rasshifr");

f.add(l1);

f.add(fl);

f.add(l2);

f.add(ps);

f.add(ch);

f.add(b);

f.add(res);

f.add(result);

f.setSize(150, 200);

f.setResizable(false);

f.setLocation(550,300);

f.setVisible(true);

b.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent event) {

if(fl.getText().length()==0){

fl.setText("test");

}

if(ps.getText().length()==0){

ps.setText("key");

}

if(ch.getSelectedItem().equals("shifr")){

String s =fl.getText();

String s1 =ps.getText();

String resy=null;

resy = crypto.shifr(s1, s);

res.setText(resy);

result.setText(ps.getText());

}

if(ch.getSelectedItem().equals("rasshifr")){

fl.setText(res.getText());

String s1 = ps.getText();

String s = fl.getText();

res.setText(crypto.rasshifr(s1, s));

}

}

});

}

});

}

}

***Класс обработки данных***

**import** java.util.Arrays;

**public** **class** crypto {

**static** String *alf*=" qwertyuiop[]asdfghjkl;'zxcvbnm,./йцукенгшщзхъфывапролджэячсмитьбю\\\"|!@#$%^&\*()\_+=-0987654321ЙЦУКЕНГШЩЗХЪФЫВАПРОЛДЖЭЯЧСМИТЬБЮQWERTYUIOPASDFGHJKLZXCVBNM";

**private** **static** **int** *n*= *alf*.length();

**static** String *matrix*[][];

**static** String shifr(String key , String text){

**int** indexkey[],indextext[],k = key.length(),t = text.length();

indexkey = **new** **int**[k];

indextext = **new** **int**[t];

*formirMatrix*();

**for**(**int** i = 0 ; i < t ; i++){

**int** j = 0;

**while**(**true**){

**if**(*alf*.charAt(j)==text.charAt(i)){

indextext[i]=j;

**break**;

}

j++;

}

}

**for**(**int** i = 0 ; i < k ; i++){

**int** j = 0;

**while**(**true**){

**if**(*alf*.charAt(j)==key.charAt(i)){

indexkey[i]=j;

**break**;

}

j++;

}

}

**int** key1 = 0;

text="";

**for**(**int** i = 0 ; i < t ; i++,key1++){

**if**(key1==k)

key1 = 0;

text+=*matrix*[indexkey[key1]][indextext[i]];

}

**return** text;

}

**static** String rasshifr(String key , String text){

**int** indexkey[],indextext[],k = key.length(),t = text.length();

indexkey = **new** **int**[t];

indextext = **new** **int**[t];

*formirMatrix*();

**int** key1 = 0;

**for**(**int** i = 0 ; i < t ; i++){

**int** j = 0;

**while**(**true**){

**if**(*alf*.charAt(j)==key.charAt(key1)){

indexkey[i]=j;

**break**;

}

j++;

}

**if**(i==k-1)

key1=0;

**else**

key1++;

}

String np[] = *perehod*(text);

**for**(**int** i = 0 ; i < t ; i++){

**int** j = 0;

**while**(**true**){

**if**(*matrix*[indexkey[i]][j].equals(np[i])){

indextext[i]=j;

**break**;

}

j++;

}

}

text ="";

**for**(**int** i = 0 ; i < t ; i++){

text+=*alf*.charAt(indextext[i]);

}

**return** text;

}

**private** **static** String [][]formirMatrix(){

*matrix* = **new** String[*n*][*n*];

String[] mat = **new** String[*n*];

mat=*perehod*(*alf*);

**for**(**int** i = 0;i < *n*; i++){

*matrix*[i]=mat;

mat=*perestanovshik*(mat);

}

**return** *matrix*;

}

**private** **static** String revers(String []text){

String textres = "";

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

textres+=text[i];

}

**return** textres;

}

**private** **static** String []perestanovshik(String []perest){

String []arg = **new** String[perest.length];

String a;

a=perest[0];

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

arg[i]=perest[i+1];

}

arg[perest.length-1]=a;

**return** arg;

}

**private** **static** String[] perehod(String got){

**int** k=got.length();

String[]faktor = **new** String[k];

**for**(**int** i=0;i<k;i++)

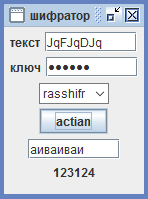
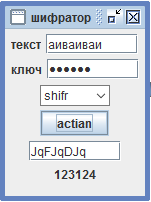
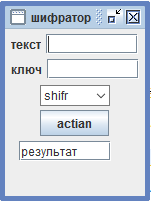
faktor[i]=got.substring(i,i+1);

**return** faktor;

}

}

**Результат работы программы:**



***Программа 9 :***

Задание : записать сериализованый класс в файл показать методы записи и чтения сериализованых данных

Блок вывода сериализованого объекта из файла

Блок ввода сериализованых объектов в файл

***Текст программы***

**package** сериализация;

**import** java.io.FileInputStream;

**import** java.io.FileOutputStream;

**import** java.io.ObjectInputStream;

**import** java.io.ObjectOutputStream;

**import** java.util.ArrayList;

**public** **class** FilesApp {

**public** **static** **void** main(String[] args) {

String filename = "test.txt";

// создадим список объектов, которые будем записывать

ArrayList<Person> people = **new** ArrayList();

Person p[]=**new** Person[3];

p[0] = **new** Person("ророш", 67, 1908., **false**);

p[1] = **new** Person("Джон", 33, 178., **true**);

p[2] = **new** Person("kolia", 21, 165., **false**);

people.add(p[0]);

people.add(p[1]);

people.add(p[2]);

**try**(ObjectOutputStream oos = **new** ObjectOutputStream(**new** FileOutputStream(filename)))

{

oos.writeObject(people);

System.***out***.println("Запись произведена");

}

**catch**(Exception ex){

System.***out***.println(ex.getMessage());

}

// десериализация в новый список

ArrayList<Person> newPeople = **null**;

**try**(ObjectInputStream ois = **new** ObjectInputStream(**new** FileInputStream(filename)))

{

newPeople=(ArrayList<Person>)ois.readObject();

}

**catch**(Exception ex){

System.***out***.println(ex.getMessage());

}

**for**(Person p1 : newPeople)

System.***out***.printf("Имя: %s \t Возраст: %d рост: %fl %b\n", p1.name, p1.age,p1.height,p1.married);

}

}

***Так данные выглядя в файле***

¬н sr java.util.ArrayListxЃТ™Зaќ I sizexp w sr ­СЃРµСЂРёР°Р»РёР·Р°С†РёСЏ.Person»)ксиЧљх I ageD heightZ marriedL namet Ljava/lang/String;xp C@ќР t

СЂРѕСЂРѕС€sq ~ !@f@ t Р”Р¶РѕРЅsq ~ @d  t koliax

Так выглядят те же данные считанные из файла при помощи метода ObjectInputStream



***Программа 10:***

Задание:

Блок вывода данных из файла с определенного байта определенное количество байт

Блок вывода всех данных из файла

Блок ввода данных в файл на определенное расстояние в байтов

Прочитать из файла задано значение прямым

Способом.

***Текст программы***

package работаСфайлом;

import java.io.File;

import java.io.IOException;

import java.io.RandomAccessFile;

class sdvig{

public static int sdvig1;

}

public class scan {

public static void main(String[] args) {

try {

String filePath = "ppp.txt";

File in = new File(filePath);

sdvig.sdvig1 = 0;

writeData(filePath, "privet", sdvig.sdvig1);

writeData(filePath, "5vasia", sdvig.sdvig1);

writeData(filePath, "kora", sdvig.sdvig1);

// writeData(filePath, "blia", 3);

// writeData(filePath, "[eq", 4);

// writeData(filePath, "privet kaka dela 14 kitaycev", 5);

// writeData(filePath, "zavtra", 6;

System.out.println(new String(readCharsFromFile(filePath, sdvig.sdvig1-13, 20)));

} catch (IOException e) {

e.printStackTrace();

}

}

// метод для записи данных в файл

private static void writeData(String filePath, String data, int seek) throws IOException {

// открываем файл с возможностью как чтения, так и записи

RandomAccessFile file = new RandomAccessFile(filePath, "rw");

// переходим на определенный индекс

file.seek(seek);

sdvig.sdvig1+=data.length()+1;

// запишем данные в этом месте

file.write(data.getBytes());

file.close();

}

// метод для чтения с файла

private static byte[] readCharsFromFile(String filePath, int seek, int chars) throws IOException {

// открываем файл только для чтения

RandomAccessFile file = new RandomAccessFile(filePath, "r");

file.seek(seek+1);

byte[] bytes = new byte[6] ;

file.read(bytes);

file.close();

return bytes;

}

}

**Результат работы программы**

