

**ФГБОУ ВПО
“МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ПУТЕЙ
СООБЩЕНИЯ”**

**Институт управления и информационных
технологий**

Кафедра “Вычислительные системы и сети”

М.Н. НИКОЛЬСКАЯ

**Рекомендовано редакционно-издательским
советом университета**

в качестве методических указаний

для студентов

**направлений «Информатика и вычислительная
техника» и «Информационная безопасность»**

МОСКВА - 2011

УДК 681.3

Н-63

**Никольская М.Н. Основы программирования
на языке Java.**

Методические указания. - М.: МИИТ, 2011, - 45 с.

В представленных методических указаниях рассматриваются основы программирования на языке Java. Изложение основ программирования представлено примерами программ, которые должны быть составлены при выполнении лабораторных работ по дисциплине «Технологии программирования».

Для студентов вузов, обучающихся по направлениям «Информатика и вычислительная техника» и «Информационная безопасность».

© ФГБ ОУ ВПО «Московский государственный
университет путей сообщения», 2011

В течение лабораторных занятий по дисциплине «Технология программирования» студент должен составить и отладить на языке JAVA 8 программ по восьми разделам:

- составление простейшей программы вычисления арифметического выражения с использованием примитивных типов данных с инициализацией исходных данных;

- составление программ обработки таблиц с использованием массивов примитивных типов данных;

- составление программ обработки массивов объектов;

- составление программ анализа классов-оболочек;

- составление программ с обработкой исключительных ситуаций, определяемых пользователем;

- составление программ для создания структур данных с использованием классов-контейнеров;

- составление программ создания файлов и их обработки файлов с использованием прямого доступа;

- составление программ с использованием средств графического интерфейса.

Далее рассматриваются примеры выполнения программ по каждому из разделов.

Задание № 1

Требуется составить программу вычисления суммы ряда .

Значок ^ означает возведения в степень.

$$S = -(x^a)/(z+2!) + (x^{(a+1)})/(2z+3!) - (x^{(a+2)})/(3z+4!) + (x^{(a+2)})/(4z+5!) - \dots$$

с заданной точностью ϵ . Исходные данные: x, z – вещественные числа, больше 1, a – целое положительное число. В случае, если абсолютное значение текущего члена ряда меньше заданной величины ϵ , цикл вычисления суммы ряда заканчивается.

```
/*основной класс pr1, где определяется основная функция
main*/
public class pr1 {
/*Исходные данные : x, a ,z ;точность e.*/
/* определение главной функции, при запуске должен
быть указан класс pr1*/
    public static void main(String[] args){
        double x=0.22,e=0.001,z=0.02;/* инициализация
исходных данных*/
        int a=5; /*инициализация исходных данных*/
        double S=0,y,x2;/*S- результат- сумма,y-текущий член
ряда,x2- рабочая переменная, соответствующая x в
текущей степени a*/
```

```

double denominator; /* текущее значение z,2z,3z и т.д.*/
int i=0,j,fact=1;/* i- текущий номер члена ряда,j- рабочая
переменная,fact-факториал*/
int k=1;/*переключатель знака*/
x2=x;
for(j=0;j<a-1;j++)/* цикл вычисления x в степени a*/
{x2=x2*x;}
do/* цикл вычисления суммы ряда*/
{denominator=(i+1)*z;/* вычисление составляющей z, 2z,
3z и т.д*/
fact=(i+2)*fact;/*вычисление факториала*/
y=x2/(denominator+fact);/* вычисление текущего члена
ряда*/
k=-k; /* определение знака для суммирования*/
System.out.println("y="+y);/* печать предыдущего
значения ряда*/
S=S+k*y; /* подсуммирование текущего члена ряда*/
x2=x2*x; /* вычисление следующего значения степени
x*/
i++; /*переход к следующему номеру члена ряда*/
}
while(y>=e); /*проверка условия выхода из цикла по
значению y- текущего члена ряда*/

```

```

    System.out.println("S="+S); /* выдача на печать значения
суммы ряда*/
    }
}

```

Контрольные вопросы к лабораторной работе № 1

1. Каковы отличия в структуре программы вычисления простого арифметического выражения на языке Си и Си++ от структуры программы на языке Java.
2. Если пакет содержит только один класс, нужно ли указывать класс, в котором находится функция main?
3. Могут ли в одном пакете определено несколько функций main и, каким образом производится запуск такого пакета?
4. Если пакет содержит несколько классов, какая информация задается при запуске такого пакета?
5. имеется ли ошибка в написании оператора `float f=2.45` и если имеется, то, как она должна быть исправлена?
6. в чем заключается отличие в использовании оператора вида `int a=23.4` в программах на языке Си и в программах на языке Java?
7. Может ли использоваться функция main без формальных параметров в Java и Си?

8. Каким образом задается фактический параметр в функции main при запуске программы на языке Java..
Каким образом в функции main распечатать значение этого параметра?

Задание № 2

Требуется составить программу, которая реализует ввод таблицы, состоящей из n строк. Каждая строка таблицы содержит следующую информацию : имя станции, время отправления поезда со станции. Требуется определить имя станции с максимальным временем отправления.

```
package mar3;
```

```
/* В данной программе информация по исходной таблице  
представлена в виде 2-х массивов:
```

```
masot[] --- массив строк из имен станций отправления
```

```
tot[] ---- массив времени отправления по каждой станции
```

```
n --- количество станций является исходным данным и  
вводится с клавиатуры
```

Алгоритм, реализованный в программе, представленной
далее:

1) ввод строки, которая определяет кол-во обрабатываемых станций, которая далее преобразуется в целое число n;

2) в цикло по i ($i=0\dots n-1$) выполнение 2-х последовательных циклов;

а) цикл ввода строки - имени станции и запись в i -ую строку массива `masot`;

б) цикл ввода строки - времени отправления и преобразования в число типа `float` и запись в i -ую строку массива `tot`;

3) цикл вычисления максимального значения из массива `tot` и определения его номера `k`;

4) выдача на печать исходной таблицы из `n` строк

5) выдача номера станции , наименования станции и времени отправления , которые соответствуют номеру `k`.
 /*
 import java.io. IOException;*/ подключение программы обработки исключительных ситуации при вводе данных*/
 public class Main{ /* основной класс*/

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

try { /*начало блока контроля исключительной ситуации при вводе*/

/*определение буфера ввода*/

StringBuffer str = new StringBuffer();

/*определение рабочей переменной char ch;*/

char ch;


```

/* n - количество строк в таблице, которое задается при
вводе*/
    int n;
/*блок ввода строки, представляющей величину n*/
    System.out.println("Введите кол-во строк в таблице");
/* цикл ввода строки, представляющей величину n; конец
строки символ \n */
    while ((ch=(char)System.in.read())!='\n')
        str.append(ch);
        n=Integer.parseInt(str.toString());
/* определение массива ссылок для n строк для станций
отправления*/
    String masot[]=new String[n];
/* определение массива из n чисел для времени
отправления*/
    float tot[]=new float[n];
/*цикл ввода строк таблицы из n строк*/
    for (int i=0;i<n;i++)
        {System.out.println(" введите имя текущей станции");
/* ввод массива символов имени станции отправления в
буфер*/
        /* сброс буфера ввода*/
        str = new StringBuffer();

```

```

while ( (ch = (char) System.in.read()) != '\n')
    str.append(ch);
/* выделение памяти для строки и запоминание имени
станции отправления в массиве masot*/
    masot[i] = new String(str.toString());
/* ввод массива символов - времени отправления*/
// сброс буфера ввода
    System.out.println(" введите время отправления с
текущей станции");
    str = new StringBuffer();
    while ( (ch = (char) System.in.read())!= '\n')
        str.append(ch);
/* преобразование строки времени в число с плавающей
точкой и запоминание в массиве tot*/
    tot[i] = Float.parseFloat(str.toString());
} /* конец цикла ввода исходных данных*/
/*вычисление максимума из массива tot ; max--
максимальное время отправления; k- номер максимума*/
    float max = tot[0]; /*max - текущий максимум*/
    int k = 0; /*номер текущего максимума*/
    for (int i = 1; i < n; i++)
    { if(max<tot[i])
        {max = tot[i]; k = i;}

```

```

    }
    /*выдача на печать исходной таблицы
    System.out.println("Ваши исходная  таблица" );
        for (int i = 0; i < n; i++)
            { System.out.println(masot[i]+" "+tot[i]); }
    /* печать результата*/
        System.out.println("ваш результат--");
        System.out.println("Станция отправления "+masot[k]+"
время отправления "+tot[k]);
    }
    /*обработка ошибки ввода-вывода*/
        catch (IOException e){
            System.out.println(e.toString());
        }
    } // конец функции main
} // конец класса Main

```

Контрольные вопросы по работе № 2

1. Опишите разницу в определении памяти для массива фиксированного размера переменных целого типа в языке Си ++ Java.
2. Какая разница в выполнении в программе на языке Си и Java оператора `a[5]=5`, если была выделена память для 4-х элементов?

3. Как измениться ввод строки из задания № 2 , представляющей целое число, если разделителем при вводе будет использоваться символ запятая или точка?
4. почему при запросе памяти для двумерного массива по оператору `new` может не указываться количество столбцов?
5. Что является признаком определения ссылки на массив примитивных данных или объектов?
6. Определить ошибки записи операторов
`int d[]=new Float[5]; int y=new int[9].`
7. Перечислите операторы перевода строк, представляющих числовые данные примитивных типов, в значения переменных этих типов.
8. В чем заключается ошибка в записи фрагмента
`int y[][]=new int[4][]; y[2][3]=2;`

Задание № 3

В этом задании требуется составить программу по условию задания № 2. Отличие состоит в том, что каждая строка исходной таблицы должна быть представлена в виде объекта. Членами данных в классе, который определяет объект, являются поля имени станции отправления и времени отправления. Исходная таблица

представляется как массив объектов. Обязательными функциям класса объекта являются функции ввода данных в объект, функция вывода данных по объекту. Доступ ко времени отправления объекта реализуется как к открытому данному через объект.

```
import java.io.IOException;
class stroka{/* определение класса для определения
объекта-строка*/
String stanotpr ;/* член данное - ссылка на объект,
соответствующий имени станции отправления*/
float otpr; /* член -данное: время отправления*/
/* метод (или член-функция) ввода данных в строку
таблицы*/
void vvod (){
    try{/* определение области контроля по вводу
данных*/
        /* ввод массива символов имени станции отправления
в буфер*/
        /* сброс буфера ввода*/
        StringBuffer str = new StringBuffer();
        char ch; /* определение рабочей переменной*/
        while ( (ch = (char) System.in.read()) != '\n')/*цикл
ввода строки до символа ent*/
```

```

        str.append(ch);
    /* запоминание станции отправления*/
        stanotpr=new String (str); /* создание объекта и
запись данных –строки*/
    /*ввод массива символов - времени отправления*/
        System.out.println(" введите время отправления с
текущей станции");
        /* сброс буфера ввода*/
        str = new StringBuffer();
        while ( (ch = (char) System.in.read())!= '\n')
            str.append(ch);
    /*преобразование строки времени в число с плавающей
точкой и запоминание*/
        otrpr= Float.parseFloat(str.toString());
    }
catch (IOException e){/* обработка ошибки при вводе*/
    System.out.println(e.toString());
    System.out.println(" ошибка при вводе информации по
станции");
    }
} // конец метода ввода строки
/* метод вывода строки*/
void    vivod() {

```

```

        System.out.println("станция "+ stanotpr+" время
отправления "+otpr); }/* конец метода вывода*/
}/* конец класса stroka*/
/*определение класса с главной функцией main*/
public class Main{
    public static void main(String[] args) {
        try {
            /*начало блока контроля исключительной ситуации при
вводе*/
            /*определение буфера ввода*/
            StringBuffer str = new StringBuffer();
            /*определение рабочей переменной*/
            char ch;
            /*n - количество строк в таблице, которое задается
при вводе*/
            int n;
            /*блок ввода строки, представляющей величину n*/
            System.out.println("Введите кол-во строк в таблице");
            /* цикл ввода строки, представляющей величину n; конец
строки символ ent*/
            while ((ch=(char)System.in.read())!='\n')
                str.append(ch);

```

```

n=Integer.parseInt(str.toString());/*преобразование
строки символов в целое число*/
/* выделение памяти для ссылок для n объектов -- для n
строк*/
    stroka stancia[]=new stroka[n];
/*цикл ввода строк таблицы из n строк*/
    for (int i=0;i<n;i++)
        {System.out.println(" данные по текущей станции");
        stancia[i]=new stroka();/* выделение памяти для
каждой станций*/
        stancia[i].vvod();/* ввод данных по станции*/
        }
/*вычисление - максимального времени отправления и
его номера*/
    float max = stancia[0].otpr; /* max - текущий максимум*/
    int k = 0; // номер текущего максимума
    for ( int i = 1; i < n; i++)
        if(max<stancia[i].otpr)
            {max = stancia[i].otpr; k = i; }
/* выдача на печать исходной таблицы*/
    System.out.println("Ваши исходные таблица" );
    for ( int i = 0; i < n; i++)
        stancia[i].vivod());

```



```

/* печать результата*/
    System.out.println("ваш результат");
    stancia[k].vivod();
}
catch (IOException e){
    System.out.println(e.toString());
}/* конец обработки ошибки ввода-вывода*/
} // конец функции main
} // конец класса Main

```

Контрольные вопросы к лабораторной работе 3

1. Почему в программе мы имеем право обращаться к времени `otpr` непосредственно через объект, а не через функцию. В чем состоит отличие от обращения на языке Си++?
2. Внесите изменения в представленную программу, при условии, что число станций постоянно и равно 10.
3. Внесите изменения в представленную программу при условии, что в строку таблицы добавляется поле времени прибытия и время представляется полями часа и минут.
4. Создайте один объект, который имеет следующую структуру: номер маршрута поезда, количество

промежуточных станций в маршруте и поля, представляющие названия станций. Определите класс, описывающий эту структуру объекта с учетом того, что количество промежуточных станций является переменной величиной и, соответственно, количество полей названия станций также переменны. Напишите функцию, которая заполняет члены-данные объекта. Каждое поле станции должны быть представлены как объект класса `String`, а все станции массивом объектов типа `String`.

5. Для указанной структуры объекта в предыдущем задании в соответствующем этому объекте классе определите 2 конструктора:

- конструктор, в котором формальные параметры задают номер маршрута и количество промежуточных станций. Названия станций определяются во время исполнения конструктора путем ввода.

- конструктор без параметров, при выполнении которого производится ввод всех членов данных объекта.

Определите два разных объекта, использующих эти конструкторы.

6. Если в классе определены несколько конструкторов, каким образом производится выбор на исполнение конкретного конструктора.

7. Дано определение 2-х объектов: `Q obj1=new Q(5,4.5); Q obj1=new Q("wwwww");`

Укажите, какое минимальное количество конструкторов должно быть определено в классе `Q` и определите наборы формальных параметров этих конструкторов.

8. Укажите, имеется ли ошибка во фрагменте `: Q obj[]=new Q[3]; obj[2].f1();`; предполагается, что в классе `Q` определена функция `f1`.

Задание №4

В четвертом задании должны быть проанализированы структура одного из системных классов и составлена программа тестирования всех констант, включенных в класс (если они имеются) и методов, принадлежащих этому классу. К этим системным классам относятся следующие: `String`, `StringBuffer`, `Character`, `Math`, `Boolean`, `Integer`, `Long`, `Float`, `Double`.

Отчет должен содержать следующую информацию:

- описание констант класса
- описание каждого конструктора, определенного в классе с указанием типа формального параметра в соответствующем конструкторе;

- описание каждого метода класса; при этом- должно быть дано описание:

заголовка метода;

-описание назначения каждого формального параметра;

- описание возвращаемой величины;

- описание исключительных ситуаций, возникающих при выполнении методов.

Кроме того, должна быть составлена программа, которая позволяет проанализировать все возможности анализируемого класса.

Пример описания и использования констант.

В классе Double определена константа,

Описание константы имеет вид :

```
public static final double MIN_VALUE
```

Константа определяет минимальное положительное число типа double, равное 2 в степени (-1074). Эта величина может быть также получена с помощью метода этого класса Double.longBitsToDouble(0x1L).

Для получения этой константы в тестирующей программе может быть записано выражение

```
double g =Double.MIN_VALUE; /*использование обращения без объекта, так как величина статическая*/
```

```
println("минимальная величина типа double равна "+g);
```

Пример описания и использования конструкторов

В классе Double определен конструктор следующего формата

```
public Double(String s)
```

Конструктор Double создает объект класса Double, в котором записывается величина типа double, которая получается путем перевода строки, представляющей десятичное число по формату плавающего типа.

Параметры конструктора

S – входной параметр, представляющий строку десятичного числа в плавающем формате, который переводится в число в формате double.

Исключительные ситуации:

Если строка символов определяет число, содержащее символы, которые не могут быть использованы в плавающем формате, то генерируется исключительная ситуация NumberFormatException (например, если имеются в строке нецифровые символы).

Для проверки реализации этого конструктора может быть записано выражение

```
Double obj=new Double(123.45);
```

Или

```
Double obj= new Double ("123.45");  
println("значение объекта равно"+obj);
```

Пример описания и использования методов

В классе Double определен метод equals

```
public boolean equals(Object obj)
```

Этот метод используется для сравнения двух объектов типа Double. Первый объект - это объект, для которого вызывается функция, второй - указан в качестве фактического параметра. При этом используются для сравнения данные типа double, записанные в как члены данные этих объектов. Результат этого метода равен true, если оба этих объекта представляют одни и те же данные.

При сравнении этих величин используется метод doubleToLongBits(double). Если этот метод использованный для величины double каждого объекта, дает одну и ту же величину типа long, то результат функции сравнения равен true.

Параметры.

Параметр задает второй объект, который участвует в сравнении.

Возвращаемая величина.

Возвращается булевский результат, значение которого равно true, если значения данных объектов равны.

Для проверки этого метода можно использовать фрагмент

```
Double obj1=new Double(1.234);
```

```
Double obj2=new Double("1.234");
```

```
boolean f=obj1.equals(obj2);
```

```
System.out.println("равенство объектов="+f);
```

Контрольные вопросы к лабораторной работе № 4

1. Преобразуйте массив символов ABCdFFh в массив символов по нижнему регистру с использованием класса String.
2. Объясните назначение функции hashCode и вычислите значение этой функции для строки "ffffff".
3. Задан фрагмент char d='F'; char g='Y'; Проанализировать определяют ли эти переменные цифры с использованием класса Character.
4. В каком случае функции классов являются статическими.
5. Преобразовать объект типа String в массив символов и определить длину этого массива символов.
6. Определить объект класса String и просуммировать все цифры этого объекта.
7. В заданном массиве символов подсчитать количество пробелов с использованием класса Character.

8. Чем отличается представление строки символов в Си и Java.

Задание № 5

В этом задании должна быть составлена программа, которая выполняет суммирование ряда, заданного в первом задании. Отличие состоит в том, что, если было задано x по абсолютному значению больше 2, то создается исключительная ситуация, . при обработке которой новое значение x получается путем вычитания из абсолютного значения x шага 0.1 до получения значения меньшего 2, и далее продолжается вычисление суммы с полученным значением x ; при этом при обработке ситуации подсчитывается количество вычитаний, полученных в цикле вычитания. Если в результате вычисления была получена отрицательная сумма генерируется исключительная ситуация того же типа, но для обработки ситуации подключается другая функция, по которой выдается значение суммы и знак текущего элемента, при которой закончился цикл вычисления с заданной точностью.

```
package javaapplication51;
```



```
class UserExcept1 extends Exception{/* определение класса
для      определения      исключительной      ситуации
пользователя*/
```

```
    UserExcept1(String g){/* конструктор класса с заданием
строки , генерируемой при генерации ситуации*/
```

```
        System.out.println(g); /*. печать имени сгенерированной
ситуации*/
```

```
    }
```

```
}/* конец класса исключительной ситуации*/
```

```
public class Main{
```

```
    public static void main(String[] args)
```

```
    {double    x=2.22,e=0.001,z=0.02;/* здание исходных
данных*/
```

```
        int a=5;
```

```
        double S=0,y,x2;
```

```
        double denominator;
```

```
        int i=0,j, fact=1;
```

```
        int k=1;
```

```
        try{/* начало контролируемой области*/
```

```
            if (Math.abs(x)>2) throw new UserExcept1("первое
исключение");/*генерация ситуации при модуле x>2*/
```

```
        }/* конец контролируемой области*/
```

```
catch(UserException obj){/* функция обработки ситуации
при модуле x больше 2*/
```

```
    System.out.println("fffff"+" "+x);
```

```
    int n=0;
```

```
    while(Math.abs(x)>2)** уменьшение модуля до
значения <2 и подсчет количества итераций*/
```

```
    { x=Math.abs(x)-0.1;n++;}
```

```
    System.out.println("fffff"+x+" "+n);
```

```
}
```

```
try{/* начало контролируемой области вычисления
суммы ряда*/
```

```
    System.out.println("aaaaaa "+x);
```

```
    x2=x;
```

```
    for(j=0;j<a-1;j++)
```

```
    {
```

```
        x2=x2*x;
```

```
    }
```

```
do {
```

```
    denominator=(i+1)*z;
```

```
    fact=(i+2)*fact;
```

```
    y=x2/(denominator+fact);
```

```
    k=-k;
```

```
    System.out.println("y="+y);
```

```

    S=S+k*y;
    x2=x2*x;
    i++;
}
while(y>=e);
System.out.println("S="+S);
if (S<0) throw new UserException("второе исключение");/*
генерация второго исключения того же типа*/
}
catch(UserException obj){ /* обработка второго
исключения*/
    System.out.println("отрицательное значени суммы" +S);
    System.out.println("значение аргумента"+x+
+"значение k="+k);
}/* конец функции catch*/
}/* конец функции main
} // конец класса

```

Контрольные вопросы к лабораторной работе № 5

1. Измените программу (исходные данные или текст программы) таким образом, чтобы не была сгенерирована 2-ая ситуация.

2. Может ли быть обработана ситуация одного и того же типа функциями `catch`, реализующими разные алгоритмы обработки?
3. Можно ли ввести новое значение величины x при генерации ситуации по первому условию вместо того, чтобы уменьшать величину x на заданный шаг?
4. Измените программу таким образом, чтобы можно было вводить шаг для получения нового значения x .
5. Напишите программу вычисления $y=(a+b)/(c-d)$, по которой при возникновении ситуации деления на нуль, выдаются значения величин c и d , производится запрос на ввод новых значений c и d , и продолжается вычисление с новыми значениями. Может быть использована системная ситуация деления на нуль.
6. Какая информация должна быть указана в заголовке функции, если ситуация создается, но не обрабатывается?

Задание № 6

При составлении этой программы необходимо создать таблицу, которая содержит строки из двух полей: поле ключа и информационное поле. Программа выполняет запись данных в две таблицы, поиск строк с заданным ключом с использованием прямого доступа, чтение

последовательности ключей и полей данных. Создание таблицы и операции чтения и поиска выполнены с использованием функций системного класса-контейнера Hashtable:

```
package hashtab;
import java.util.Hashtable;
import java.util.Enumeration;
public class Demohash{
    public static void main(String[] args) throws Exception {
        /*создание объекта класса Hashtable, который будет
        содержать ключи-объекты класса String и объекты класса
        Double*/
        Hashtable doubleTable = new Hashtable();
        /*занесение в таблицу нескольких данных
        примитивного типа с ключами; значение примитивов
        должны быть представлены объектами*/
        doubleTable.put ("2.5", new Double(3788)) ;/* запись строк
        в объект */
        doubleTable.put("2.539", new Double(37.88));
        doubleTable.put("2.537", new Double(788));
        doubleTable.put("2.535", new Double(4));
        /* поиск по ключу объекта, объект должен быть приведен
        по типу Double*/
```

```

Double doubleObject = (Double) doubleTable.get("2.537");
    if (doubleObject != null)
        System.out.println("Найдено значение =" +
doubleObject.toString());
/*Создание 2- го объекта класса Hashtable из номеров и
имен*/
    Hashtable tablename = new Hashtable(100);
/*занесение нескольких записей*/
/*строки-литералы являются объектами, поэтому они
непосредственно используются как ключи и как элементы
данных*/
    tablename.put("1000", "andropov");
    tablename.put("1003", "nicolaev");
    tablename.put("1001", "ivanov");
    tablename.put("1010", "pichugin");
    String ObjectStr = (String) tablename.get("1010");
    if (ObjectStr != null)
        {System.out.println("Найдено значение =" +
ObjectStr.toString());}
        else {System.out.println("запись отсутствует"); };
/*Вывод ключей таблицы*/
    System.out.println("\n Ключи базы данных");

```

```

/*создание объекта keyEnum типа Enumeration , в который
записываются все ключи всех записей, занесенных в
таблицу; метод key принадлежит интерфейсу
Enumeration*/

/*в цикле просматриваются последовательно все ключи
таблицы*/

Enumeration keyEnum=(Enumeration)tablename.keys();
while (keyEnum.hasMoreElements() ) {
    String key = (String) keyEnum.nextElement();
    System.out.println(key);
}

/* Вывод имен; создание объекта nameEnum типа
Enumeration , который будет содержать все
информационные поля записей , занесенных в объект
класса Hashtable; метод elements принадлежит
интерфейсу Enumeration; в цикле просматриваются
последовательно все элементы таблицы*/

System.out.println("\n Имена");
Enumeration nameEnum = (Enumeration)
tablename.elements();
while (nameEnum.hasMoreElements() ) {
    String name = (String) nameEnum.nextElement();
    System.out.println(name);
}

```

```

/*Вывод ключей и имен путем поиска*/
System.out.println("\n Ключи и Имена");
keyEnum = (Enumeration) tablename.keys();
while (keyEnum.hasMoreElements()) {
    String key = (String) keyEnum.nextElement();
    String name = (String) tablename.get(key);
    if (name == null)
        throw new Exception("Ошибка хеш=таблицы");
    System.out.println(key + ";" + name);
} } }

```

результаты

Найдено значение =788.0

Найдено значение =pichugin

Ключи базы данных

1010

1003

1001

1000

Имена

pichugin

nicolaev

ivanov

andronov

Ключи и Имена

1010:pichugin

1003;nicolaev

1001;ivanov

1000;andronov

Контрольные вопросы к работе № 6

1. Можно ли в качестве ключей использовать переменные целого типа?
2. Можно ли записать в объект класса `Hashtable` записать строку вида “`dddd`”, 234?
3. Как определить количество записей в объекте класса `Hashtable`?
4. Чем отличаются функции `elements` и `keys`?
5. Какая функция используется при определении адреса записи строки в объект класса `Hashtable`?
6. Совпадает ли порядок записи строк в таблицу с порядком выдачи этих записей путем последовательного просмотра? Если нет, то почему?
7. Какая функция должна быть использована при получении результата поиска?

8. Измените программу таким образом, чтобы можно было обработать ситуацию Exception и выдать сообщение о том, что записи с заданным ключом нет.

Задание № 7

В этом задании создается файл, в который записываются последовательно десятичные числа с использованием системных классов, и производится чтение записи с заданным номером с использованием средств прямого доступа.

```
package prf5;
import java.io.*;
public class Main{
    /* функция ввода числа*/
    static double getchislo ()throws IOException{/* функция
получения исходных данных*/
        StringBuffer buffer = new StringBuffer();
        char ch;
        System.out.println("введите действительное десятичное
число ");
        while ( (ch = (char) System.in.read()) != '\n')
            buffer.append(ch);
        return new Double (buffer.toString()).doubleValue();
```

```

/* возврат переменной типа double*/
} /* конец ввода числа и возврат величины типа double*/
/*главная программа*/
public static void main(String args[]){
int size =3; /*количество чисел в создаваемом файле*/
try {
    // создание файловых объектов
    FileOutputStream f=new FileOutputStream("C:/\"mmm");
    BufferedOutputStream b=new BufferedOutputStream(f);
    DataOutputStream fl float = new DataOutputStream(b);
    /* Первый модуль реализует следующий алгоритм
создания файла
1. первой записью по оператору fl float.writeInt;(size);
    заносится целое число, которое определяет число
    записываемых чисел по оператору
    fl float.writeDouble.
2 . производится запись массива чисел по оператору
    fl float.writeDouble(getchislo() ); функция getchislo()
определяет число типа double, которые выводится в файл;
используется типизированный вывод в файл числа типа
double , поэтому используется writeDouble*/
fl float.writeInt;(size);

```

```

for (int i = 0; i < size; i++)/* цикл записи в файл по
типизированному выводу чисел с плавающей точкой*/
    fl float.writeDouble(getchislo());
int r=fl float.size();/*вычисление размера созданного
файла в байтах*/
    System.out.println(r+" байт было записано");
    fl float.close();
}
catch (IOException e) {
    System.err.println(e.toString());
}
} /*конец главной функции*/
}

```

Результаты:

введите действительное десятичное число

234.6

введите действительное десятичное число

45.7

введите действительное десятичное число

34.7

28 байт было записано

/* количество байтов было подсчитано как $4+8*3$ */

/*2-ой модуль выполняет чтение чисел с указанным номером из созданного файла*/

```
import java.io.*;
```

```
public class readfile {
```

```
    // функция ввода номера записи
```

```
    public static int getNumberRecord() throws IOException {
```

```
        StringBuffer buffer = new StringBuffer();
```

```
        char ch;
```

```
        System.out.println(" введите номер записи");
```

```
        while ( (ch = (char) System.in.read()) != '\n')
```

```
            buffer.append(ch);
```

```
        return new Integer(buffer.toString()).intValue();
```

```
        // возврат переменной целого типа
```

```
    } /*конец ввода имени и возврат объект типа String*/
```

/*главная программа чтения записи с заданным номером*/

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

```
    int size; /*количество чисел в файле, которое записано в файле как первое число*/
```

```
    int номер; // номер записи
```

```
    double value; // значение записи
```

```
    int sizeOfInt=4; /*размер переменной типа int*/
```

```

int sizeOfDouble=8; /* размер переменной типа double*/
boolean vixod=false; // указатель выхода
try{
    // создание файловых объектов
    File f1= new File("C:/mmm ");
    RandomAccessFile f2= new RandomAccessFile (f1,"r");
    size=f2.readInt(); /*чтение количества чисел в файле*/
    System.out.println("\n в файле записано " + size +
"чисел");
    while (!vixod){
//чтение номера искомой записи
        nomer=getNumberRecord();
        vixod=(nomer==-1);
        if(!vixod) {
            /*вычисление номера байта числа, которое
необходимо прочитать и переход к этому числу*/
            f2.seek(sizeOfInt + ((nomer-1) * sizeOfDouble));
            value=f2.readDouble();
            System.out.println("Запись с номером "+ nomer+"=" +
value); }
            System.out.println("вывод окончен "); }
    } catch (IOException e){
        System.err.println(e.toString()+" fff");
    }
}

```

```

    }
} // конец функции main()
}

```

Контрольные вопросы к лабораторной работе №7

1. Исполните указанную программу с получением ошибки вывода при получении записи с заданным номером.
2. Измените эти программы для записи последовательности из пар чисел двух типов данных: float, int . Количество чисел задается величиной n.
3. Измените в исходной программе оператор чтения записи с номером n на оператор чтения readFloat и объясните результат, который будет получен.
4. Создайте файл, который содержит n записей , каждая из которых состоит из двух полей: тип float и тип double. Прочитайте запись с заданным номером.

Задание № 8

В этом задании должна быть составлена программа, которая обеспечивает создание графического пользовательского интерфейса .

В этой программе реализуется следующий алгоритм:

- вводится некоторый текст;
- введенный текст меняется путем прибавления к каждому введенному символу текста некоторой константы (выполняется кодирование текста).
- полученный текст преобразуется в исходный текст путем вычитания из каждого символа той же константы, которая была использована при кодировании (то есть выполняется декодирование текста)

В программе кодирования и декодирования текста используется следующая структура данных:

- 2 поля редактирования:

поле Redak1, в которое записывается исходный и декодируемый текст

поле Redak2, в которое записывается закодированный текст

- 3 кнопки:

кнопVvod – нажатие кнопки вызывает функцию кодирования текста из поля Redak1 и запись в поле Redak2 закодированного текста

кнопVivod - нажатие кнопки вызывает функцию декодирования текста из поля Redak2 и запись в поле Redak1 декодированного текста

-knopSbros- нажатие кнопки приводит к стиранию полей Redak1 и Redak2

- 2 метки с названием кнопок polText1 и polText2, расположенных рядом с кнопками

Для создания указанных полей используются соответствующие классы графических библиотек.

Управляющие элементы расположены в контейнере Pole1.

```

/*использование графического режима*/
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
class Koder implements ActionListener { /*подключение
интерфейса */
JTextField Redak1; /*определение 2-х текстовых полей*/
JTextField Redak2; *
Koder() { //конструктор
JFrame Pole1=new JFrame("Pervii primer"); /*создание
нового контейнера*/
/* установка диспетчера компоновки для определения
схемы размещения полей*/
Pole1.getContentPane().setLayout(new FlowLayout());
// установка размера контейнера
Pole1.setSize(340,200);

```

```
/*определение функции завершения при закрытии
программы*/
Pole1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
//определение 2-х меток
JLabel poltext1=new JLabel("pole vvoda");
JLabel poltext2=new JLabel("pole kodirovania");
/*определение 2-х текстовых полей редактирования для
записи данных*/
Redak1=new JTextField(40);
Redak2=new JTextField(40);
/* установка команд действия для полей редактирования*/
Redak1.setActionCommand("KODIROVANIE");
Redak2.setActionCommand("DEKODIROVANIE");
/*связывание обработчиков событий с полями
редактирования*/
Redak1.addActionListener(this);
Redak2.addActionListener(this);
/**добавление полей редактирования в контейнер Pole1*/
Pole1.getContentPane().add(poltext1);/* добавление метки*/
Pole1.getContentPane().add((Redak1); /*добавление поля
редактирования*/
Pole1.getContentPane().add(poltext2);
Pole1.getContentPane().add(Redak2);
```

```

//создание 3-х кнопок
JButton knopVvod=new JButton("KODIROVANIE");
JButton knopVivod=new JButton("DEKODIROVANIE");
JButton knopSbros=new JButton("Sbros");
/*связывание кнопок с обработчиками события типа
Action*/
knopVvod.addActionListener(this);
knopVivod.addActionListener(this);
knopSbros.addActionListener(this);
/*добавление в контейнер Pole1 кнопок*/
Pole1.getContentPane().add((knopVvod));
Pole1.getContentPane().add((knopVivod));
Pole1.getContentPane().add((knopSbros));
// визуализация контейнера Pole1
Pole1.setVisible(true);
} // конец конструктора
/* определение функции обработки событий */
public void actionPerformed(ActionEvent sob){
    If (sob.getActionCommand().equals("KODIROVANIE") ){
/*если выбрана функция кодирования, чтение и
кодирования текста из поля Redak1*/
        StringBuffer str=new StringBuffer(Redak1.getText());
        for(int i=0;i<str.length();i++)

```

```

        str.setCharAt(i,(char)(str.charAt(i)+3));
        Redak2.setText(str.toString());/*запись закодированного
текста*/
    }
    else /* декодирование поля Redak2*/
    if (sob.getActionCommand().equals("DEKODIROVANIE"))
    {StringBuffer str=new StringBuffer(Redak2.getText());
        for(int i=0;i<str.length(); i++)
            str.setCharAt(i,(char)(str.charAt(i)-3));
        Redak1.setText(str.toString());
    }
    else {// сброс 2-х полей
        Redak1.setText("");
        Redak2.setText("");
    }
    public static void main(String args[]){
    // создание кода в потоке обработки событий
        SwingUtilities.invokeLater(new Runnable(){
    public void run(){
        New Koder();// запуск программы
    }});}}

```

Контрольные вопросы к лабораторной работе № 8

1. Объясните, каким образом производится инициализация исполнения программы.
2. Измените программу при изменении имен функций обработки нажатий кнопок на следующие имена:– функция1, функция 2, функция3.
3. Измените размер полей редактирования.
4. Измените функцию очистки полей исходных данных на функцию заполнения этих полей объединением строк в этих полях
5. Перечислите классы, которые используются в программе.
6. Объясните назначение предложения `implements ActionListener`.
7. Отключите в программе исполнение функции кодирования, не удаляя полей редактирования .

Учебно-методическое издание

Никольская Марина Николаевна

Основы программирования на языке Java

Методические указания

Подписано в печать	Формат 60x84/16	Тираж 100 экз
--------------------	-----------------	---------------

Изд. N 152-11	Усл. печ. л.	Заказ N
---------------	--------------	---------

150048, г. Ярославль, Московский пр-т, д. 151.
Типография Ярославского филиала МИИТ.