

Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Ярославский государственный технический университет»
Кафедра «Информационные системы и технологии»

Контрольная работа защищена
с оценкой _____
Преподаватель
канд. техн. наук, доцент
_____ В. К. Маевский
« ____ » _____ 2023

КОНТРОЛЬНАЯ РАБОТА

По дисциплине
«Информационные технологии»

ЯГТУ 09.03.02 — 011 К/Р

Работу выполнил
студент гр. ДЦИС-27
_____ П.Л. Конторин
« 27 » _____ апреля 2023

Оглавление

Оглавление	2
Лабораторная работа №1	3
1.1	3
1.2	4
Лабораторная работа №2	5
2.1	5
2.2	9
Лабораторная работа №3	10
Лабораторная работа №4	13
4.1	13
4.2	15
4.3	17
4.4	20
Лабораторная работа №5	24
5.1	24
5.2	25
5.3	26
5.4	27
5.5	29
5.6	31
Лабораторная работа №6	33
6.1.1	33
6.1.2	35
6.2	37
6.3	39
6.4	42
6.5	44
Лабораторная работа №7	47
Лабораторная работа №8	51
Лабораторная работа №9	58
9.1	58
9.2	61
9.3	65

Лабораторная работа №1

1.1

Составить программу вычисления и печати таблицы значений функции при заданных значениях параметров:

$a=2,7$; $b=-0,27$; $0 \leq x \leq 7$; $\Delta x=0,5$;
 $y=(a+b)/(e^x + \cos x)$, если $x < 2,3$;
 $y=(a+b)(x+1)$, если $2,3 \leq x < 5$;
 $y=e^x + \sin x$, если $x \geq 5$.

```
public class Main {  
    public static void main(String[] args) {  
        double a=2.7, b=-0.27, x, y;  
        System.out.println("x\ty");  
        for (x=0; x<=7; x+=0.5)  
        {  
            System.out.print(x+"\t");  
            if (x<2.3) System.out.println(y=(a+b)/(Math.exp(x)+Math.cos(x)));  
            else if (x<5) System.out.println(y=(a+b)*(x+1));  
            else System.out.println(y=Math.exp(x)+Math.sin(x));  
        }  
    }  
}
```

Output - JavaApplication (run)

```
init:
deps-jar:
Compiling 1 source file to C:\Users\33319\JavaApplication\build\classes
compile:
run:
x      y
0.0    1.215
0.5    0.9618795525114057
1.0    0.7457226512586986
1.5    0.5337812970070043
2.0    0.3484915561871757
2.5    8.505
3.0    9.72
3.5    10.935
4.0    12.15
4.5    13.365
5.0    147.45423482791347
5.5    243.98639193865
6.0    403.14937799453617
6.5    665.3567530324497
7.0    1097.2901450271772
BUILD SUCCESSFUL (total time: 0 seconds)
```

1.2

Составить программу вычисления и печати таблицы значений функции, которая является суммой бесконечного ряда.

Для заданных значений аргумента X вычислить значения суммы S и функции Y или Z . Вычисление S произвести с точностью E .

$$S = (4x/\pi) \sum_{k=1}^{\infty} 1/((2k-1)^2 + x^2); z = \text{th}(\pi x/2); -\pi/4 \leq x \leq \pi/4; \Delta x = 0.05\pi; \\ E = 10^{-3}$$

```
public class Main {
    public static void main(String[] args) {
        double x, S, z, T, E=0.001;
        int k;
        System.out.println("\tx\t\t\tS\t\t\tz");
        for (x=-Math.PI/4; x<=Math.PI/4; x+=0.05*Math.PI)
        {
            System.out.print(x+"\t");
            T=4*x/(Math.PI+Math.PI*x*x);
            k=1;
            S=0;
            while (Math.abs(T)>E)
```

```

    {
        S+=T;
        k++;
        T=4*x/(Math.PI*Math.pow((2*k-1),2)+Math.PI*x*x);
    }
    if (S==0) System.out.print("\t\t");
    System.out.print(S+"\t");
    if (S==0) System.out.print("\t\t");
    System.out.println(z=Math.tanh(Math.PI*x/2));
}
}
}

```

```

Output - JavaApplication (run)

init:
deps-jar:
Compiling 1 source file to C:\Users\33319\JavaApplication\build\classes
compile:
run:

      x                S                z
-0.7853981633974483    -0.8280325758070912    -0.8436493662488775
-0.6283185307179586    -0.7417860021789444    -0.756063262355724
-0.47123889803846897    -0.6167870779679284    -0.6292782611896978
-0.3141592653589793    -0.44698339161734146    -0.45697426895333493
-0.15707963267948966    -0.23472138161430955    -0.24185187974379493
0.0                    0.0                    0.0
0.15707963267948966    0.23472138161430955    0.24185187974379493
0.3141592653589793    0.44698339161734146    0.45697426895333493
0.47123889803846897    0.6167870779679284    0.6292782611896978
0.6283185307179586    0.7417860021789444    0.756063262355724
0.7853981633974483    0.8280325758070912    0.8436493662488775
BUILD SUCCESSFUL (total time: 0 seconds)

```

Лабораторная работа №2

2.1

Даны два одномерных целочисленных массива. Произвести поиск заданного значения в первом из них – методом последовательного перебора с использованием барьерного элемента, а во втором – бинарный поиск, предварительно отсортировав этот массив методом вставки. Первый массив отсортировать затем выбором наименьшего элемента. Произвести слияние полученных массивов.

В одномерном массиве, состоящем из n вещественных элементов, вычислить:

- 1) номер минимального по модулю элемента массива;
- 2) сумму модулей элементов массива, расположенных после первого

отрицательного элемента.

```
class Main{
public static void main(String[] args){
    int a[]=new int [11];
    int b[]=new int [10];
    double c[]=new double [10];
    double cmin, sumao=0;
    int i, j, imin=0, isumao=0;
    // Заполнение массива a случайными числами:
    System.out.println("Массив a:");
    for (i=0;i<10;i++){
        a[i]=(int)(200*Math.random()-100);
        System.out.print(a[i]+" ");
    }
    System.out.println();
    /* Поиск заданного значения методом последовательного перебора
    с использованием барьерного элемента:*/
    int x=14; // Заданное для поиска значение
    for(i=0,a[10]=x;a[i]!=x;i++){
        if (i==10) System.out.println("Искомое число "+x+" в массиве нет.");
        else System.out.println("Искомое число "+x+" в массиве есть, его индекс: "+i);
    }
    // Сортировка массива a выбором наименьшего элемента:
    i=0;
    while (i<9){
        int min=i;
        j=i+1;
        while (j<10){
            if (a[j]<a[min]) min=j;
            j++;
        }
        int buf=a[i];
        a[i]=a[min];
        a[min]=buf;
        i++;
    }
    System.out.println("Отсортированный массив a:");
    for (i=0;i<10;i++){
        System.out.print(a[i]+" ");
    }
    System.out.println("\n");

    // Заполнение массива b случайными числами:
    System.out.println("Массив b:");
    for (i=0;i<10;i++){
```

```

        b[i]=(int)(200*Math.random()-100);
        System.out.print(b[i]+" ");
    }
    System.out.println();
    // Сортировка массива b методом вставки:
    for (i=1;i<10;i++){
        int buf=b[i];
        for (j=i;j>0&&buf<b[j-1];j--){
            b[j]=b[j-1];
        }
        b[j]=buf;
    }
    System.out.println("Отсортированный массив b:");
    for (i=0;i<10;i++)
        System.out.print(b[i]+" ");
    System.out.println();
    // Бинарный поиск:
    x=14; // Заданное для поиска значение
    int left=0;
    int right=b.length-1;
    int middle=(left+right)/2;
    while (left<=right&&b[middle]!=x){
        if (x>b[middle]) left=middle+1;
        else right = middle - 1;
        middle=(left+right)/2;
    }
    if (left>right) System.out.println("Искомое число "+x+" в массиве нет.\n");
    else System.out.println("Искомое число "+x+" в массиве есть, его индекс:
"+middle+"\n");

    // Слияние массивов a и b
    System.out.println("Слияние массивов a и b:");
    int ab[]=new int [20];
    int ia, ib, iab;
    for (ia=ib=iab=0;iab<20;iab++){
        if ( ia < 10 ){
            if ( ib < 10 ){
                if (a[ia]<b[ib])
                    ab[iab]=a[ia++];
                else ab[iab]=b[ib++];
            }
            else ab[iab]=a[ia++];
        }
        else ab[iab]=b[ib++];
    }
}

```

```

for (i=0;i<20;i++)
    System.out.print(ab[i]+" ");
System.out.println("\n");

System.out.println("Массив с:");
// Заполнение массива с случайными числами:
for (i=0;i<10;i++){
    c[i]=(double)(200*Math.random()-100);
    c[i]=Math.round(c[i]*100.0)/100.0;
    System.out.print(c[i]+" ");
}
// Поиск индекса минимального по модулю элемента:
cmin=c[0];
for(i=0;i<10;i++){
    if (Math.abs(c[i])<Math.abs(cmin)){
        imin=i;
        cmin=c[i];
    }
}
System.out.println();
System.out.println("Номер минимального по модулю элемента массива с:
"+(imin+1));
// Поиск индекса элемента, расположенного после первого отрицательного
элемента:
for(i=0;i<10;i++){
    if (c[i]<0) {
        isumao=i+1;
        break;
    }
}
// Подсчет суммы модулей элементов, расположенных после первого
отрицательного элемента:
if (isumao==0) System.out.println("В массиве с отрицательных элементов нет.");
else{
    for(i=isumao;i<10;i++) sumao+=Math.abs(c[i]);
    System.out.print("Сумма модулей элементов массива с, расположенных
после первого отрицательного элемента: ");
    System.out.println(String.format("%.2f", sumao));
}
}
}

```


Output - JavaApplication (run)

```
init:
deps-jar:
Compiling 1 source file to C:\Users\33319\JavaApplication\build\classes
compile:
run:
Массив a:
26 70 -58 -18 -30 38 -51 98 42 30
Искомое число 14 в массиве нет.
Отсортированный массив a:
-58 -51 -30 -18 26 30 38 42 70 98

Массив b:
95 -45 71 54 22 59 93 -70 -68 14
Отсортированный массив b:
-70 -68 -45 14 22 54 59 71 93 95
Искомое число 14 в массиве есть, его индекс: 3

Слияние массивов a и b:
-70 -68 -58 -51 -45 -30 -18 14 22 26 30 38 42 54 59 70 71 93 95 98

Массив c:
33.78 13.61 -26.81 95.2 54.96 12.25 21.33 9.44 95.87 -60.92
Номер минимального по модулю элемента массива c: 8
Сумма модулей элементов массива c, расположенных после первого отрицательного элемента: 349,97
BUILD SUCCESSFUL (total time: 0 seconds)
```

2.2

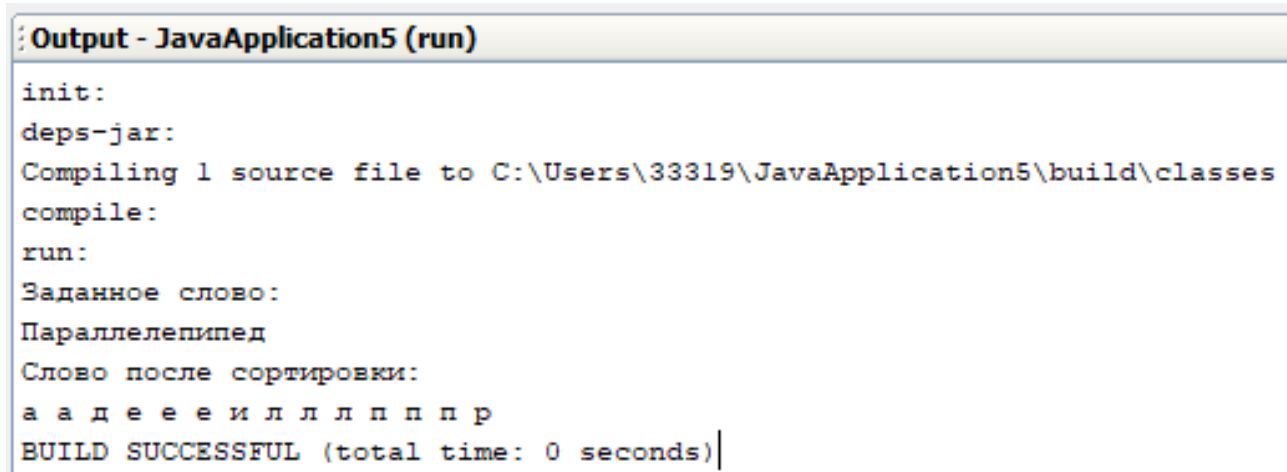
Дано слово. Произвести сортировку данного слова методом пузырька с использованием индекса.

```
class Main{
public static void main(String[] args){
    char[] word=new char[] {'П','а','р','а','л','л','е','л','е','п','и','п','е','д'};
    int n=word.length;
    int[] index=new int[n]; // Массив индексов
    int i,j,f,buf;
    for (i=0;i<n;i++) // Заполнение массива индексов
        index [i]=i;
    System.out.println("Заданное слово:");
    System.out.println(word);
    word[0]=Character.toLowerCase(word[0]);
    for (i=1;i<n;i++){ //Сортировка
        f=0;
        for (j=0;j<n-1;j++){
            if (word[index[j]]>word[index[j+1]]){
                buf=index[j];
                index[j]=index[j+1];
                index[j+1]=buf;
                f=1;
            }
        }
    }
}
```

```

        if (f==0) break;
    }
    System.out.println("Слово после сортировки:");
    for (i=0;i<n;i++)
        System.out.print(word[index[i]]+" ");
    }
}

```



Output - JavaApplication5 (run)

```

init:
deps-jar:
Compiling 1 source file to C:\Users\33319\JavaApplication5\build\classes
compile:
run:
Заданное слово:
Параллелепипед
Слово после сортировки:
а а д е е е и л л л п п п р
BUILD SUCCESSFUL (total time: 0 seconds)

```

Лабораторная работа №3

Дана целочисленная прямоугольная матрица, все элементы которой различны. Поменять местами столбцы, содержащие минимальный и максимальный элементы матрицы. Найти количество строк, среднее арифметическое элементов которых меньше заданной величины.

```

class Main{
public static void main(String[] args){
    int i, j, k, p, q, m=10, n=10;
    int[][] mas=new int[m][n];
    /* Заполнение массива случайными числами
       с проверкой на повторяемость*/
    for (i=0;i<m;i++){
        for (j=0;j<n;j++){
            mas[i][j]=(int)(200*Math.random()-100);
            for (p=0;p<m;p++){
                for (q=0;q<n;q++){
                    if (p>=i&&q>=j) break;
                    if(mas[i][j]==mas[p][q]){
                        if (j>0) j--;
                        else {i--; j=n-1;}
                        break;
                    }
                }
            }
        }
    }
}

```

```

    }
  }
}
System.out.println("Заданная матрица:");
for (i=0;i<m;i++){
  for (j=0,k=0;j<n;j++,k++){
    System.out.print(mas[i][j]+(k!=n-1?"\t":"\n"));
  }
}

```

/* Поиск индексов столбцов, содержащих минимальный и максимальный элементы.

Подсчет количества строк, среднее арифметическое элементов которых меньше заданной величины.*/

```

int max=mas[0][0];
int min=mas[0][0];
int jmax=0, jmin=0, sum;
int x=4; // Заданная величина
double aver;
int kaver=0;
for (i=0;i<m;i++){
  sum=0;
  k=0;
  for (j=0;j<n;j++){
    if (mas[i][j]>max){
      max=mas[i][j];
      jmax=j;
    }
    if (mas[i][j]<min){
      min=mas[i][j];
      jmin=j;
    }
    sum+=mas[i][j];
  }
  aver=(double)sum/m;
  if (aver<x) kaver++;
}

```

```

System.out.println();

```

// Перестановка столбцов:

if(jmax==jmin) System.out.println("Минимальный и максимальный элементы содержатся в одном столбце.");

```

else{
  int[]buf=new int[m];
  for (i=0;i<m;i++)
    buf[i]=mas[i][jmax];
}

```

```

        for (i=0;i<m;i++)
            mas[i][jmax]=mas[i][jmin];
        for (i=0;i<m;i++)
            mas[i][jmin]=buf[i];
        System.out.println("Матрица после перестановки столбцов, имеющих
максимальный и минимальный элементы:");
        for (i=0;i<m;i++){
            for (j=0,k=0;j<n;j++,k++){
                System.out.print(mas[i][j]+(k!=n-1?"\t":"\n"));
            }
        }
        System.out.println();
        System.out.println("Количество строк, среднее арифметическое элементов
которых меньше "+x+": "+kaver);
    }
}

```

Output

Debugger Console ×

JavaApplication5 (run) ×

```

init:
deps-jar:
Compiling 1 source file to C:\Users\33319\JavaApplication5\build\classes
compile:
run:
Заданная матрица:
9      -12    91    -29    -4    -96    -62    -92    -77    80
-79    -83    75    40    4    -81    -69    76    23    -32
-89    70    -90    -15    -84    -55    -70    -30    -65    5
-47    56    -60    -46    -58    57    -93    -95    -88    10
38    18    -66    33    -52    37    -20    -42    -35    -78
-44    54    64    14    71    63    -17    30    -82    -14
92    61    -5    42    62    -98    19    -61    3    72
95    99    58    -2    -6    15    46    -41    20    -37
44    -45    82    6    -36    45    32    -8    96    -85
-25    0    -38    -43    -18    60    26    86    89    -54

Матрица после перестановки столбцов, имеющих максимальный и минимальный элементы:
9      -96    91    -29    -4    -12    -62    -92    -77    80
-79    -81    75    40    4    -83    -69    76    23    -32
-89    -55    -90    -15    -84    70    -70    -30    -65    5
-47    57    -60    -46    -58    56    -93    -95    -88    10
38    37    -66    33    -52    18    -20    -42    -35    -78
-44    63    64    14    71    54    -17    30    -82    -14
92    -98    -5    42    62    61    19    -61    3    72
95    15    58    -2    -6    99    46    -41    20    -37
44    45    82    6    -36    -45    32    -8    96    -85
-25    60    -38    -43    -18    0    26    86    89    -54

Количество строк, среднее арифметическое элементов которых меньше 4: 5
BUILD SUCCESSFUL (total time: 0 seconds)

```

Лабораторная работа №4

1. Освоить графический интерфейс среды NetBeans.
2. Создать в среде NetBeans приложения по лабораторным работам: №1, №2 (задание I, только свой вариант), №3 с использованием графического интерфейса.

4.1

Составить программу вычисления и печати таблицы значений функции при заданных значениях параметров:

$a=2,7$; $b=-0,27$; $0 \leq x \leq 7$; $\Delta x=0,5$
 $y=(a+b)/(e^x + \cos x)$, если $x < 2,3$;
 $y=(a+b)(x+1)$, если $2,3 \leq x < 5$;
 $y=e^x + \sin x$, если $x \geq 5$.

```
public class NewJFrame extends javax.swing.JFrame {

    public NewJFrame() {
        initComponents();
    }

    private void aboutMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
        JOptionPane.showMessageDialog(null, "Работу выполнил студент группы
ДЦИС-27 Конторин П.Л.");
    }

    private void cleanMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
        int i;
        for (i=0; i<15; i++)
            jTable1.setValueAt("", i, 0);
        for (i=0; i<15; i++)
            jTable1.setValueAt("", i, 1);
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        double a=2.7, b=-0.27, x, y;
        double xn = Double.parseDouble(jTextField3.getText());
        double xk = Double.parseDouble(jTextField2.getText());
        double dx = Double.parseDouble(jTextField1.getText());
        int i=0;
        for (x=xn; x<=xk; x+=dx)
        {
            jTable1.setValueAt(String.valueOf(x), i, 0);
            if
```

(x<2.3)

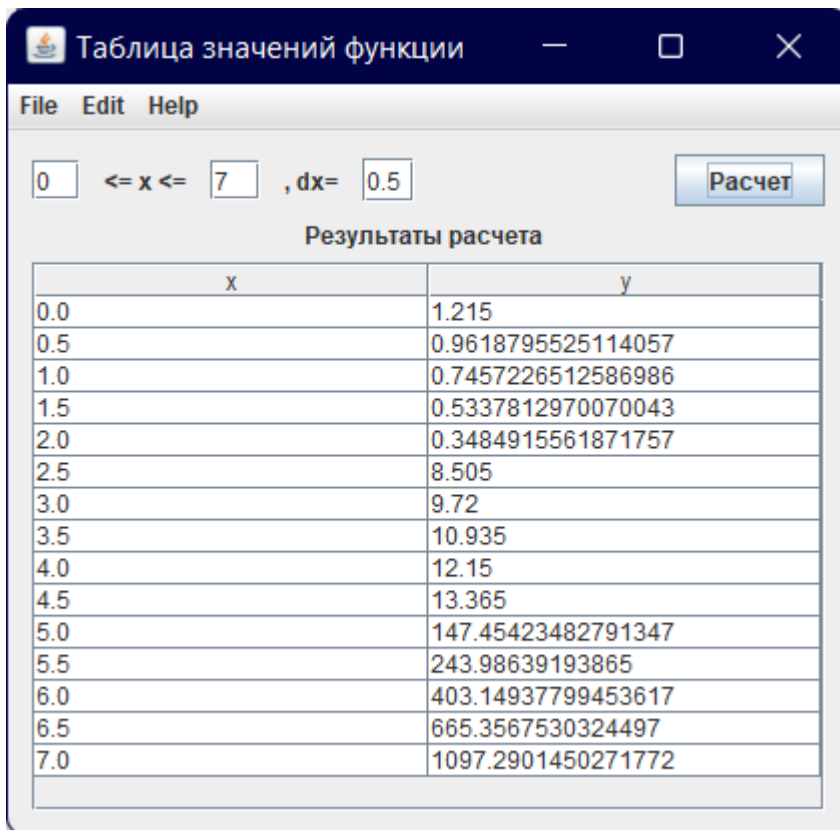
```

jTable1.setValueAt(String.valueOf(y=(a+b)/(Math.exp(x)+Math.cos(x))), i, 1);
    else if (x<5) jTable1.setValueAt(String.valueOf(y=(a+b)*(x+1)), i, 1);
    else jTable1.setValueAt(String.valueOf(y=Math.exp(x)+Math.sin(x)), i, 1);
    i++;
}
}

private void closeMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            //new NewJFrame().setVisible(true);
            NewJFrame frame = new NewJFrame();
            frame.setVisible(true);
            frame.setLocationRelativeTo(null);
        }
    });
}
}

```



x	y
0.0	1.215
0.5	0.9618795525114057
1.0	0.7457226512586986
1.5	0.5337812970070043
2.0	0.3484915561871757
2.5	8.505
3.0	9.72
3.5	10.935
4.0	12.15
4.5	13.365
5.0	147.45423482791347
5.5	243.98639193865
6.0	403.14937799453617
6.5	665.3567530324497
7.0	1097.2901450271772

4.2

Составить программу вычисления и печати таблицы значений функции, которая является суммой бесконечного ряда.

Для заданных значений аргумента X вычислить значения суммы S и функции Y или Z . Вычисление S произвести с точностью E .

$$S = (4x/\pi) \sum_{k=1}^{\infty} 1/((2k-1)^2 + x^2); \quad z = \text{th}(\pi x/2); \quad -\pi/4 \leq x \leq \pi/4; \quad \Delta x = 0.05\pi; \\ E = 10^{-3}$$

```
public class NewJFrame extends javax.swing.JFrame {
    static double xn=-Math.PI/4;
    static double xk=Math.PI/4;
    static double xd=0.05*Math.PI;

    public NewJFrame() {
        initComponents();
    }

    private void aboutMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
        JOptionPane.showMessageDialog(null,"Работу выполнил студент группы
ДЦИС-27 Конторин П.Л.");
    }

    private void defaultButtonActionPerformed(java.awt.event.ActionEvent evt) {
        xn=-Math.PI/4;
        xnLabel.setText("-П/4");
        jTextField1.setText("");
        xk=Math.PI/4;
        xkLabel.setText("П/4");
        jTextField2.setText("");
        xd=0.05*Math.PI;
        xdLabel.setText("0.05П");
        jTextField3.setText("");
    }

    private void setButtonActionPerformed(java.awt.event.ActionEvent evt) {
        if
(jTextField1.getText().equals("")||jTextField2.getText().equals("")||jTextField3.getTex
t().equals(""));
        else{
            xn=Double.parseDouble(jTextField1.getText());
            xnLabel.setText(jTextField1.getText());
            xk=Double.parseDouble(jTextField2.getText());
            xkLabel.setText(jTextField2.getText());
```

```

        xd=Double.parseDouble(jTextField3.getText());
        xdLabel.setText(jTextField3.getText());
    }
}

private void cleanMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    for (int i=0;i<20;i++)
        for (int j=0;j<3;j++)
            jTable1.setValueAt("", i, j);
}

private void calculateButtonActionPerformed(java.awt.event.ActionEvent evt) {
    double x, S, z, T, E=0.001;
    int i, j, k;
    for (i=0;i<20;i++)
        for (j=0;j<3;j++)
            jTable1.setValueAt("", i, j);
    i=0;
    try{
        for (x=xn; x<=xk; x+=xd)
        {
            jTable1.setValueAt(String.valueOf(x), i, 0);
            T=4*x/(Math.PI+Math.PI*x*x);
            k=1;
            S=0;
            while (Math.abs(T)>E)
            {
                S+=T;
                k++;
                T=4*x/(Math.PI*Math.pow((2*k-1),2)+Math.PI*x*x);
            }
            jTable1.setValueAt(String.valueOf(S), i, 1);
            jTable1.setValueAt(String.valueOf(z=Math.tanh(Math.PI*x/2)), i, 2);
            i++;
        }
    }
    catch(ArrayIndexOutOfBoundsException e){
        JOptionPane.showMessageDialog(null,
            "The results of calculations should be less than 20 iterations.");
    }
}

```

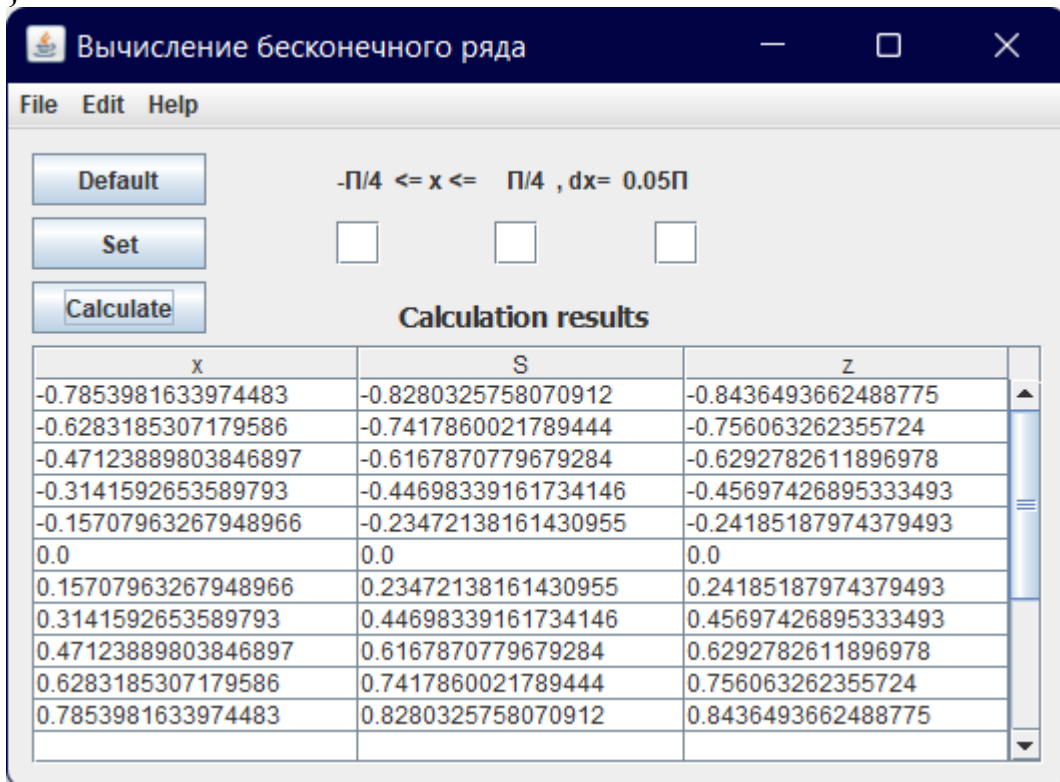


```

private void closeMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            //new NewJFrame().setVisible(true);
            NewJFrame frame = new NewJFrame();
            frame.setVisible(true);
            frame.setLocationRelativeTo(null);
        }
    });
}
}

```



4.3

Даны два одномерных целочисленных массива. Произвести поиск заданного значения в первом из них – методом последовательного перебора с использованием барьерного элемента, а во втором – бинарный поиск, предварительно отсортировав этот массив методом вставки. Первый массив отсортировать затем выбором наименьшего элемента. Произвести слияние полученных массивов.

В одномерном массиве, состоящем из n вещественных элементов, вычислить:

- 1) номер минимального по модулю элемента массива;
- 2) сумму модулей элементов массива, расположенных после первого

отрицательного элемента.

```
public class LR_4_3 extends javax.swing.JFrame {

    public LR_4_3() {
        initComponents();
    }

    private void oПрограммеMenuItemActionPerformed(java.awt.event.ActionEvent
evt) {
        jOptionPane1.showMessageDialog(null,"Работу выполнил студент группы
ДЦИС-27 Конторин П.Л.");
    }

    private void очиститьMenuItemActionPerformed(java.awt.event.ActionEvent evt)
{
        jLabel2.setText("");
        jLabel4.setText("");
        jLabel8.setText("");
        jLabel9.setText("");
    }

    private void createButtonActionPerformed(java.awt.event.ActionEvent evt) {
        int n=Integer.parseInt(jTextField1.getText());
        double a[]=new double [n];
        int i, imin=0, isumao=0;
        double sumao=0;
        jLabel5.setText("Одномерный массив, состоящий из "+n+" вещественных
элементов:");
        jLabel2.setText("");
        jLabel9.setText("");
        // Заполнение массива с случайными числами:
        for (i=0;i<n;i++){
            a[i]=(double)(200*Math.random()-100);
            a[i]=Math.round(a[i]*100.0)/100.0;
            jLabel2.setText(jLabel2.getText()+" "+Double.toString(a[i]));
        }
        // Поиск индекса минимального по модулю элемента:
        double amin=a[0];
        for(i=0;i<n;i++){
            if (Math.abs(a[i])<Math.abs(amin)){
                imin=i;
                amin=a[i];
            }
        }
    }
}
```

```

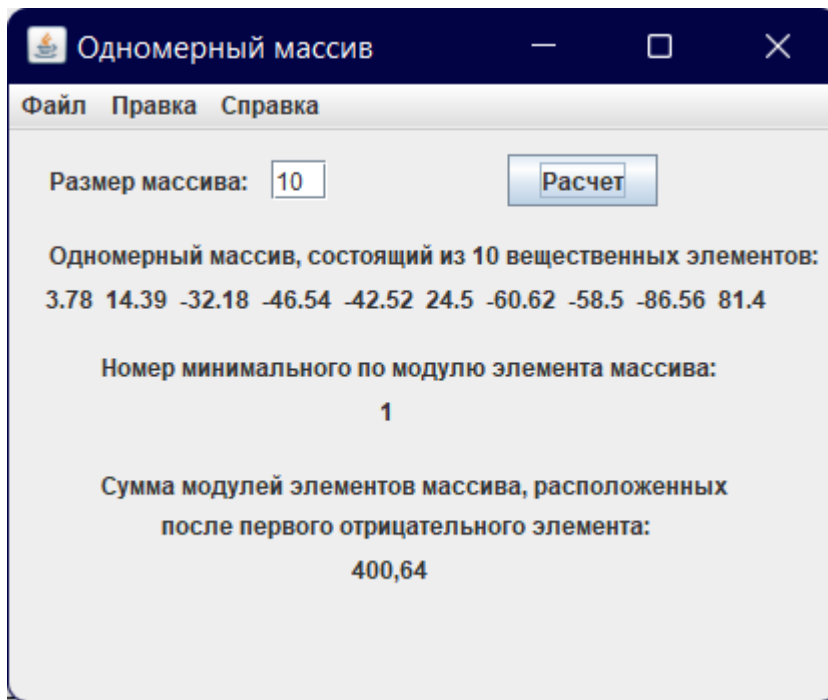
    imin++;
    jLabel4.setText(""+imin);//проверить +
    // Поиск индекса элемента, расположенного после первого отрицательного
элемента:
    for(i=0;i<n;i++){
        if (a[i]<0) {
            isumao=i+1;
            break;
        }
    }
    // Подсчет суммы модулей элементов, расположенных после первого
отрицательного элемента:
    if (isumao==0){
        jLabel8.setText("");
        jLabel9.setText("В массиве отрицательных элементов нет.");
    }
    else for(i=isumao;i<n;i++){
        sumao+=Math.abs(a[i]);
        jLabel8.setText(String.format("%.2f", sumao));
    }

}

private void exitMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            LR_4_3 frame = new LR_4_3();
            frame.setVisible(true);
            frame.setLocationRelativeTo(null);
        }
    });
}
}

```



4.4

Дана целочисленная прямоугольная матрица, все элементы которой различны. Поменять местами столбцы, содержащие минимальный и максимальный элементы матрицы. Найти количество строк, среднее арифметическое элементов которых меньше заданной величины.

```
public class LR_4_4 extends javax.swing.JFrame {

    public LR_4_4() {
        initComponents();
    }

    private void oПрограммеMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
        JOptionPane.showMessageDialog(null, "Работу выполнил студент группы ДЦИС-27 Конторин П.Л.");
    }

    private void очиситьMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
        jTextArea1.setText("");
        jTextArea2.setText("");
        jLabel7.setText("");
    }

    private void расчетButtonActionPerformed(java.awt.event.ActionEvent evt) {
        int i, j, k, p, q;
        int m=0, n=0, d=0;
```

```

int m1=Integer.parseInt(jTextField1.getText());
int n1=Integer.parseInt(jTextField2.getText());
int d1=Integer.parseInt(jTextField3.getText());
if (m1*n1>=d1)
    JOptionPane.showMessageDialog(null,"<html><p align=center>Условие
неповторимости элементов матрицы не может быть соблюдено." +
        "<p align=center>Измените значения входных параметров.");
else{
    n=n1; m=m1; d=d1;
    int[][] mas=new int[m][n];
    jTextField1.setText("");
    jTextField2.setText("");
    //Заполнение массива случайными числами с проверкой на
повторяемость:
    for (i=0;i<m;i++){
        for (j=0;j<n;j++){
            mas[i][j]=(int)(d*Math.random()-d/2);
            for (p=0;p<m;p++){
                for (q=0;q<n;q++){
                    if (p>=i&&q>=j) break;
                    if(mas[i][j]==mas[p][q]){
                        if (j>0) j--;
                        else {i--; j=n-1;}
                        break;
                    }
                }
            }
        }
    }
    for (i=0;i<m;i++){
        for (j=0,k=0;j<n;j++,k++){
            jTextField1.setText(jTextField1.getText()+mas[i][j]+(k!=n-1?"\t":"\n"));
        }
    }
    /** Поиск индексов столбцов, содержащих минимальный и
максимальный элементы.
    * Подсчет количества строк, среднее арифметическое элементов которых
меньше заданной величины.
    */
    int max=mas[0][0];
    int min=mas[0][0];
    int jmax=0, jmin=0, sum;
    int x=Integer.parseInt(jTextField4.getText()); // Заданная величина
    double aver;
    int kaver=0;

```

```

for (i=0;i<m;i++){
    sum=0;
    k=0;
    for (j=0;j<n;j++){
        if (mas[i][j]>max){
            max=mas[i][j];
            jmax=j;
        }
        if (mas[i][j]<min){
            min=mas[i][j];
            jmin=j;
        }
        sum+=mas[i][j];
    }
    aver=(double)sum/m;
    if (aver<x) kaver++;
}
// Перестановка столбцов:
if(jmax==jmin)  jTextArea2.setText("Минимальный   и   максимальный
элементы содержатся в одном столбце.");
else{
    int[]buf=new int[m];
    for (i=0;i<m;i++)
        buf[i]=mas[i][jmax];
    for (i=0;i<m;i++)
        mas[i][jmax]=mas[i][jmin];
    for (i=0;i<m;i++)
        mas[i][jmin]=buf[i];
    for (i=0;i<m;i++){
        for (j=0,k=0;j<n;j++,k++){
            jTextArea2.setText(jTextArea2.getText()+mas[i][j]+(k!=n-
1?"\t":"\n"));
        }
    }
    jLabel7.setText("Количество строк, среднее арифметическое элементов
которых меньше "+x+": "+kaver);
}
}

private void выходMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}

public static void main(String args[]) {

```

```

java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        LR_4_4 frame = new LR_4_4();
        frame.setVisible(true);
        frame.setLocationRelativeTo(null);
    }
});
}
}

```

Прямоугольная матрица

Файл Правка Помощь

Количество строк: Количество столбцов: Диапазон чисел: Заданная величина:

Заданная матрица:

-71	-96	-5	16	56	4	30	-76	67	-2
34	-33	-12	50	92	-61	20	-69	-82	37
42	-13	-60	-93	-30	53	-9	-38	-20	-7
8	-10	52	-78	-47	-37	-24	36	-4	49
-54	57	26	45	70	55	-46	-49	-91	-42
-83	28	12	-52	29	54	3	19	15	-16
-86	32	-35	-80	82	-79	-41	-73	-1	2
22	68	-44	18	-39	40	-14	31	-84	65
-98	7	-45	58	-72	-53	79	87	-87	-88
90	24	-29	71	78	88	-74	-55	89	-85

Матрица после перестановки столбцов, имеющих максимальный и минимальный элементы:

56	-96	-5	16	-71	4	30	-76	67	-2
92	-33	-12	50	34	-61	20	-69	-82	37
-30	-13	-60	-93	42	53	-9	-38	-20	-7
-47	-10	52	-78	8	-37	-24	36	-4	49
70	57	26	45	-54	55	-46	-49	-91	-42
29	28	12	-52	-83	54	3	19	15	-16
82	32	-35	-80	-86	-79	-41	-73	-1	2
-39	68	-44	18	22	40	-14	31	-84	65
-72	7	-45	58	-98	-53	79	87	-87	-88
78	24	-29	71	90	88	-74	-55	89	-85

Количество строк, среднее арифметическое элементов которых меньше 4: 8

Лабораторная работа №5

5.1

Ввести с клавиатуры слово и символ. Произвести циклический сдвиг символов слова на один заданный символ вправо и присоединить этот символ к концу слова. Вывести слово и символ. Создать консольное приложение с использованием класса String.

```
package com.mycompany.lr_5_1;

import java.io.*;

public class LR_5_1 {
    public static void main(String[] args) throws IOException {
        String s;
        char c;
        BufferedReader br=new BufferedReader(new
        InputStreamReader(System.in,"Cp1251"));
        System.out.print("Введите слово: ");
        s=br.readLine();
        System.out.print("Введите символ: ");
        c=(char)br.read();
        int len = s.length();
        char mas[] = new char[len+1];
        s.getChars(0,len,mas,0);
        char buf = mas[len-1];
        for (int i=len-1;i>0;i--)
            mas[i]=mas[i-1];
        mas[0]=buf;
        mas[len]=c;
        System.out.println("Результат: " + String.valueOf(mas));
    }
}
```



```
Output - Run (LR_5_1) x
-----< com.mycompany:LR_5_1 >-----
Building LR_5_1 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.1.0:exec (default-cli) @ LR_5_1 ---
Введите слово: Параллелепипед
Введите символ: Q
Результат: дПараллелепипеQ

BUILD SUCCESS

Total time: 32.068 s
Finished at: 2023-03-30T11:18:39+03:00
-----
```

5.2

Ввести с клавиатуры слово и символ. Произвести циклический сдвиг символов слова на один заданный символ вправо и присоединить этот символ к концу слова. Вывести слово и символ. Создать консольное приложение с использованием класса StringBuffer.

```
package com.mycompany.lr_5_2;
```

```
import java.util.*;
public class LR_5_2 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in,"Cp1251");
        System.out.print("Введите слово: ");
        String s=in.next();
        System.out.print("Введите символ: ");
        String c=in.next();
        StringBuffer s1 = new StringBuffer();
        s1.append(s);
        int len = s1.length();
        char buf = s1.charAt(len-1);
        for (int i=len-1;i>0;i--)
            s1.setCharAt(i, s1.charAt(i-1));
        s1.setCharAt(0, buf);
        System.out.println(s1+c);
    }
}
```

```

Output x
Run (LR_5_2) x Run (LR_5_2) x
-----< com.mycompany:LR_5_2 >-----
Building LR_5_2 1.0-SNAPSHOT
-----[ jar ]-----
--- exec-maven-plugin:3.1.0:exec (default-cli) @ LR_5_2 ---
Введите слово: Параллелепипед
Введите символ: Q
дПараллелепипеQ
BUILD SUCCESS
Total time: 38.372 s
Finished at: 2023-03-31T12:20:27+03:00

```

5.3

Ввести с клавиатуры слово и символ. Произвести циклический сдвиг символов слова на один заданный символ вправо и присоединить этот символ к концу слова. Вывести слово и символ. Создать приложение с использованием графического интерфейса.

```

import javax.swing.JOptionPane;
public class LR_5_3 extends javax.swing.JFrame {
    public LR_5_3() {
        initComponents();
    }

    private void выходMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
        System.exit(0);
    }

    private void оПрограммеMenuItemActionPerformed(java.awt.event.ActionEvent
evt) {
        JOptionPane.showMessageDialog(null, "Работу выполнил студент группы
ДЦИС-27 Конторин П.Л.");
    }

    private void результатButtonActionPerformed(java.awt.event.ActionEvent evt) {
        String s=jTextField1.getText();
        String c=jTextField2.getText();
        StringBuffer s1 = new StringBuffer();
        s1.append(s);
        int len = s1.length();
        char buf = s1.charAt(len-1);

```

```

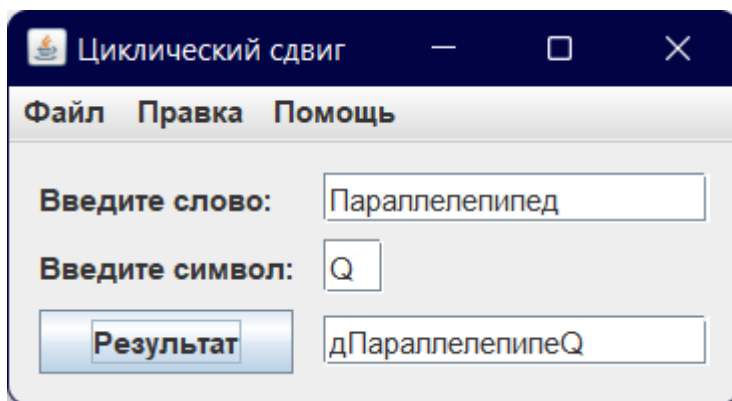
        for (int i=len-1;i>0;i--)
            s1.setCharAt(i, s1.charAt(i-1));
        s1.setCharAt(0, buf);
        jTextField3.setText(s1+c);
    }

    private void очиститьMenuItemActionPerformed(java.awt.event.ActionEvent evt)
    {
        jTextField1.setText("");
        jTextField2.setText("");
        jTextField3.setText("");
    }

    public static void main(String args[]) {

        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                LR_5_3 frame = new LR_5_3();
                frame.setVisible(true);
                frame.setLocationRelativeTo(null);
            }
        });
    }
}

```



5.4

Написать программу, которая считывает с клавиатуры текст и выводит на экран только строки, не содержащие двузначных чисел. Создать консольное приложение с использованием класса String.

```

package ystu.lr_5_4;
import java.io.*;
public class LR_5_4 {
    public static void main(String[] args) throws IOException {

```

```

BufferedReader br = new BufferedReader(new
InputStreamReader(System.in,"Cp1251"));
System.out.print("Введите количество строк: ");
int k = Integer.parseInt(br.readLine());
System.out.println("Введите текст из " + k + " строк:");
String [] text = new String [k];
for (int i=0;i<k;i++)
    text [i]= br.readLine();
System.out.println("\nСтроки, не содержащие двузначных чисел:");
int l = 0;
for (int i=0;i<k;i++){
    text[i] = 'Q'+text[i];
    char row [] = new char [text[i].length()+1];
    text[i].getChars(0, text[i].length(), row, 0);
    int print = 1;
    for (int j=0;j<text[i].length();j++){
        if (Character.isDigit(row[j])==false&&Character.isDigit(row[j+1])&&
            Character.isDigit(row[j+2])&&Character.isDigit(row[j+3])==false){
            print = 0;
            break;}
    }
    if (print==1){
        System.out.println(text[i].substring(1) );
        l++;
    }
}
if (l==0) System.out.println("строк, не содержащих двузначных чисел, в
тексте нет.");
}
}

```

```
Output - Run (LR_5_4) x
-----< YSTU:LR_5_4 >-----
Building LR_5_4 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.1.0:exec (default-cli) @ LR_5_4 ---
Введите количество строк: 6
Введите текст из 6 строк:
Около 123 значений из 5 примеров;
14 заданий с 3 переменными.
Константа Wgh34irnb!
581321 5482 510 8513 1
Параметр может принимать 420 значений.
Ближе всего по значению число 25?

Строки, не содержащие двузначных чисел:
Около 123 значений из 5 примеров;
581321 5482 510 8513 1
Параметр может принимать 420 значений.

BUILD SUCCESS

Total time: 14.027 s
Finished at: 2023-04-04T23:37:17+03:00
-----
```

5.5

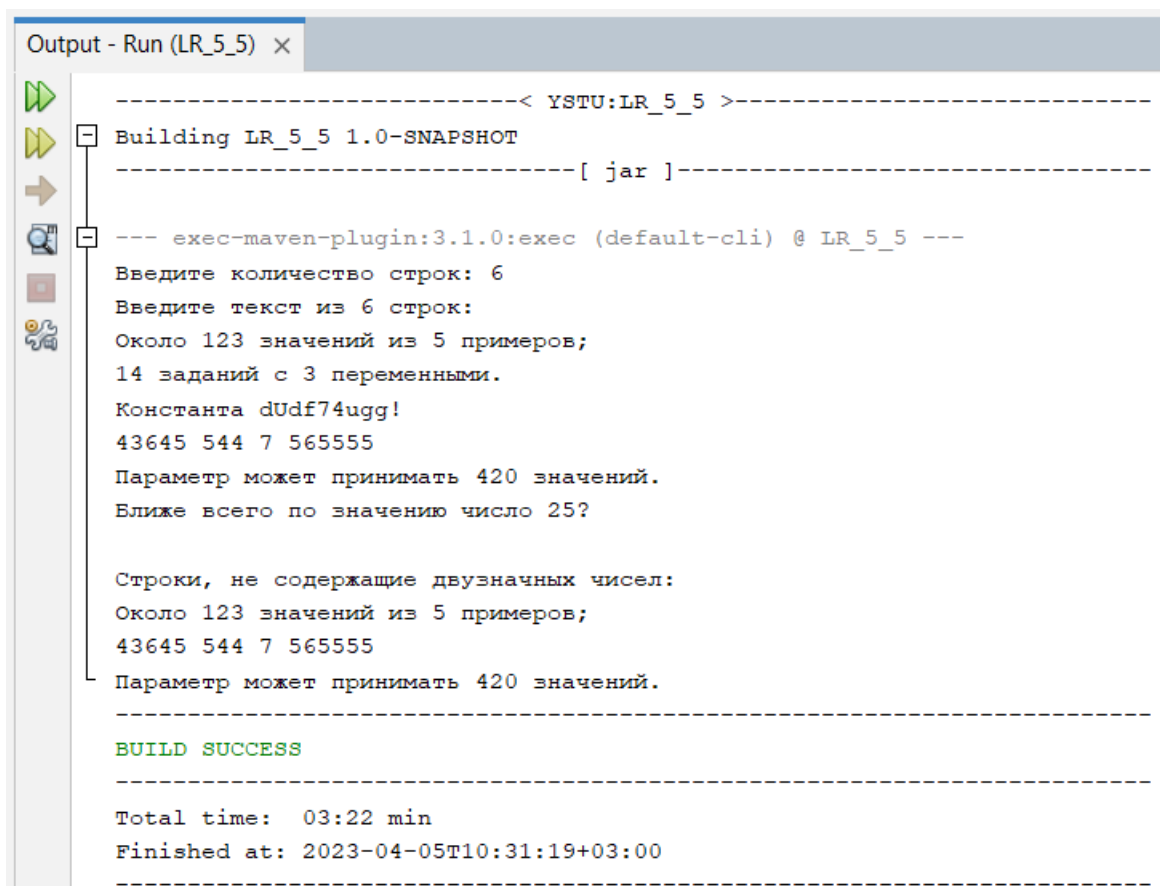
Написать программу, которая считывает с клавиатуры текст и выводит на экран только строки, не содержащие двузначных чисел. Создать консольное приложение с использованием класса StringBuffer.

```
package ystu.lr_5_5;
import java.util.*;
public class LR_5_5 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in,"Cp1251");
        System.out.print("Введите количество строк: ");
        int k = Integer.parseInt(in.nextLine());
        System.out.println("Введите текст из " + k + " строк:");
        String [] text = new String [k];
        for (int i=0;i<k;i++)
            text [i]= in.nextLine();
        System.out.println("\nСтроки, не содержащие двузначных чисел:");
        int l = 0;
        for (int i=0;i<k;i++){
            StringBuffer row = new StringBuffer ();
            row.append('Q').append(text[i]).append('Q');
            int print = 1;
            for (int j=0;j<row.length()-1;j++){
```

```

        if
(Character.isDigit(row.charAt(j))==false&&Character.isDigit(row.charAt(j+1))&&
Character.isDigit(row.charAt(j+2))&&Character.isDigit(row.charAt(j+3))==false){
            print = 0;
            break;}
    }
    if (print==1){
        row.deleteCharAt(row.length()-1);
        System.out.println(row.deleteCharAt(0) );
        l++;
    }
}
if (l==0) System.out.println("строк, не содержащих двузначных чисел, в
тексте нет.");
}
}

```



```

Output - Run (LR_5_5) x
-----< YSTU:LR_5_5 >-----
Building LR_5_5 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.1.0:exec (default-cli) @ LR_5_5 ---
Введите количество строк: 6
Введите текст из 6 строк:
Около 123 значений из 5 примеров;
14 заданий с 3 переменными.
Константа dUdf74ugg!
43645 544 7 565555
Параметр может принимать 420 значений.
Ближе всего по значению число 25?

Строки, не содержащие двузначных чисел:
Около 123 значений из 5 примеров;
43645 544 7 565555
Параметр может принимать 420 значений.

BUILD SUCCESS

Total time: 03:22 min
Finished at: 2023-04-05T10:31:19+03:00

```

5.6

Написать программу, которая считывает с клавиатуры текст и выводит на экран только строки, не содержащие двузначных чисел. Создать приложение с использованием графического интерфейса.

```
import javax.swing.JOptionPane;
public class LR_5_6 extends javax.swing.JFrame {
    public LR_5_6() {
        initComponents();
    }
    private void выходMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
        System.exit(0);
    }

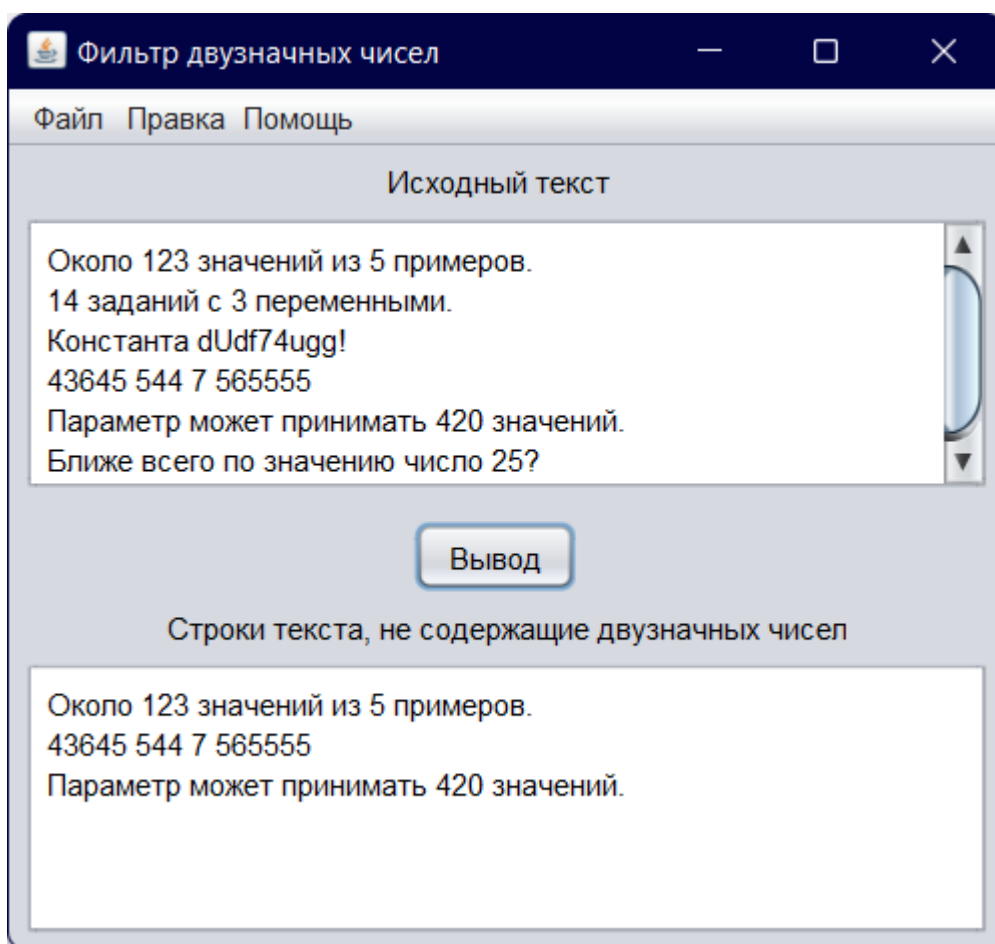
    private void oПрограммеMenuItemActionPerformed(java.awt.event.ActionEvent
evt) {
        JOptionPane.showMessageDialog(null, "<html><p align=center>Лабораторная
работа 5.6"
            + "<p align=center>Выполнил студент группы ДЦИС-27" + "<p
align=center>Конторин П.Л.");
    }

    private void выводButtonActionPerformed(java.awt.event.ActionEvent evt) {
        String [] text = jTextArea1.getText().split("\n");
        int l = 0;
        for (int i=0;i<text.length;i++){
            StringBuffer row = new StringBuffer();
            row.append('Q').append(text[i]).append('Q');
            int print = 1;
            for (int j=0;j<row.length()-1;j++){
                if
(Character.isDigit(row.charAt(j))==false&&Character.isDigit(row.charAt(j+1))&&
Character.isDigit(row.charAt(j+2))&&Character.isDigit(row.charAt(j+3))==false){
                    print = 0;
                    break;}
            }
            if (print==1){
                row.deleteCharAt(row.length()-1);
                jTextArea2.setText(jTextArea2.getText()+row.deleteCharAt(0)+"\n");
                l++;
            }
        }
        if (l==0) jTextArea2.setText("Строк, не содержащих двузначных чисел, в
тексте нет.");
    }
}
```

```

private void очиститьMenuItemActionPerformed(java.awt.event.ActionEvent evt)
{
    jTextArea1.setText("");
    jTextArea2.setText("");
}
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            //new LR_5_6().setVisible(true);
            LR_5_6 frame = new LR_5_6();
            frame.setVisible(true);
            frame.setLocationRelativeTo(null);
        }
    });
}
}

```



Лабораторная работа №6

6.1.1

Создать метод, который определяет, входит ли список L1 в список L2.

```
package ystu.lr_6_1_1;
class ListElement {
    ListElement next;
    int data;
}
class List {
    private ListElement head;
    private ListElement tail;

    void addBack(int data) {
        ListElement a = new ListElement();
        a.data = data;
        if (head == null){
            head = a;
            tail = a;
        }
        else {
            tail.next = a;
            tail = a;
        }
    }

    void checkList(List l){
        ListElement a = head;
        ListElement b;
        int i = 0;
        while (a != null){
            i++;
            b = l.head;
            while (b != null){
                if (a.data == b.data)
                    i--;
                b = b.next;
            }
            a = a.next;
        }
        if (i == 0)
            System.out.println("Список L1 входит в список L2");
        else
```

```

        System.out.println("Список L1 не входит в список L2");
    }

    void printList(){
        ListElement t = head;
        while (t != null){
            System.out.print(t.data + " ");
            t = t.next;
        }
    }
}

```

```

public class LR_6_1_1 {

    public static void main(String[] args) {
        // Формирование L1:
        List L1 = new List();
        for (int i=3;i<8;i++)
            L1.addBack(i);
        // Формирование L2:
        List L2 = new List();
        for (int i=1;i<11;i++)
            L2.addBack(i);
        // Печать L1:
        System.out.println("Список L1:");
        L1.printList();
        System.out.println();
        // Печать L2:
        System.out.println("Список L2:");
        L2.printList();
        System.out.println();
        // Проверка вхождения L1 в L2:
        L1.checkList(L2);
    }
}

```

```
Output - Run (LR_6_1_1) ×
-----< YSTU:LR_6_1_1 >-----
Building LR_6_1_1 1.0-SNAPSHOT
-----[ jar ]-----
--- exec-maven-plugin:3.1.0:exec (default-cli) @ LR_6_1_1 ---
Список L1:
3 4 5 6 7
Список L2:
1 2 3 4 5 6 7 8 9 10
Список L1 входит в список L2
-----
BUILD SUCCESS
-----
Total time: 0.326 s
Finished at: 2023-04-06T12:15:19+03:00
-----
```

6.1.2

Создать метод, который вставляет в непустой список L, элементы которого упорядочены по неубыванию, новый элемент E так, чтобы сохранилась упорядоченность ($t=\text{float}$).

```
package ystu.lr_6_1_2;
class ListElement {
    ListElement next;
    float data;
}
class List {
    private ListElement head;
    private ListElement tail;

    // Функция формирования упорядоченного списка:
    void addSort(float data){
        ListElement a = new ListElement();
        a.data = data;
        if (head == null){
            head = a;
            tail = a;
        }
        else{
            ListElement i = head;
            ListElement j = head;
            while (i != null){
                if (data < i.data){
                    a.next = i;
                    if (i == head){
```

```

        head = a;
    }
    else{
        j.next = a;
    }
    return;
}
if(i == head)
    i = i.next;
else{
    i = i.next;
    j = j.next;
}
}
a.next = null;
tail.next = a;
tail = a;
}
}

```

```

void printList(){
    ListElement a = head;
    while (a != null){
        System.out.print(a.data + " ");
        a = a.next;
    }
}

```

```

}
public class LR_6_1_2 {

    public static void main(String[] args) {
        float a[]=new float [10];
        for (int i=0;i<10;i++){
            a[i]=(float)(2000*Math.random()-1000);
            a[i]=(float)(Math.round(a[i]*100.0)/100.0);
        }
        List L = new List();
        for (int i=0;i<10;i++)
            L.addSort(a[i]);
        System.out.println("Список L:");
        L.printList();
        float E = (float)(2000*Math.random()-1000);
        E = (float)(Math.round(E*100.0)/100.0);
        System.out.println("\n\nЭлемент E: " + E + "\n");
    }
}

```

```

    L.addSort(E);
    System.out.println("Список L с элементом E:");
    L.printList();
}
}

```

```

Output - Run (LR_6_1_2) x
-----< YSTU:LR_6_1_2 >-----
Building LR_6_1_2 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.1.0:exec (default-cli) @ LR_6_1_2 ---
Список L:
-939.37  -768.42  -494.59  -478.98  101.47  300.12  565.3  893.29  896.05  935.35

Элемент E: -391.69

Список L с элементом E:
-939.37  -768.42  -494.59  -478.98  -391.69  101.47  300.12  565.3  893.29  896.05  935.35
-----
BUILD SUCCESS
-----
Total time:  0.328 s
Finished at: 2023-04-06T20:44:32+03:00
-----

```

6.2

Создать метод, который определяет, есть ли в списке L хотя бы два одинаковых элемента E.

```

package ystu.lr_6_2;
import java.util.Random;

```

```

class Item{
    char symbol;
    Item next;
    Item prev;
}
class List {
    private Item head;
    private Item tail;

    void createList(char data) {
        Item a = new Item();
        a.symbol = data;
        if (head == null){
            head = a;
            tail = a;
        }
    }
}

```

```

        else {
            tail.next = a;
            a.prev = tail;
            tail = a;
        }
        tail.next = head;
        head.prev = tail;
    }

    void seekTwo () {
        Item a = head;
        while (a != tail) {
            Item b = a.next;
            while (b != tail) {
                if (a.symbol == b.symbol) {
                    System.out.println("\nВ списке есть хотя бы два одинаковых
элемента.");
                    return;
                }
                b = b.next;
            }
            if (a.symbol == tail.symbol) {
                System.out.println("\nВ списке есть хотя бы два одинаковых
элемента.");
                return;
            }
            a = a.next;
        }
        System.out.println("\nВ списке нет одинаковых элементов.");
    }

    void printList() {
        Item a = head;
        while (a != tail) {
            System.out.print(a.symbol + " ");
            a = a.next;
        }
        System.out.print(tail.symbol);
    }
}

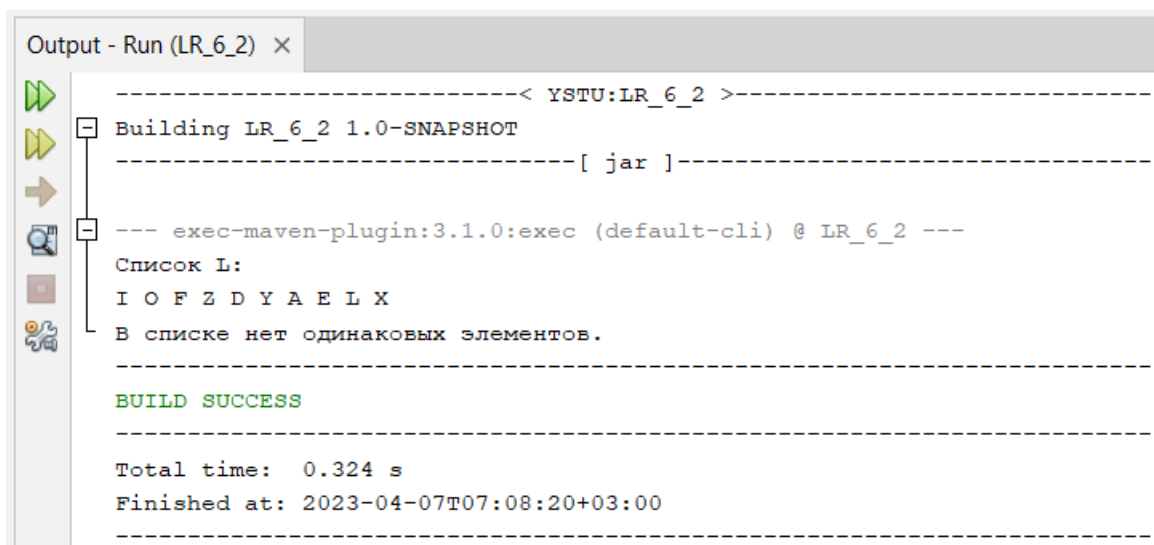
public class LR_6_2 {
    public static void main(String[] args) {
        // Формирование массива случайных символов:
        char a[]=new char [10];
    }
}

```

```

Random random = new Random();
for (int i=0;i<10;i++){
    int a1 = random.nextInt(25 + 1) + 65;
    a[i] = (char)a1;
}
List L = new List();
for (int i=0;i<10;i++)
L.createList(a[i]);
System.out.println("Список L:");
L.printList();
L.seekTwo();
}
}

```



```

Output - Run (LR_6_2) x
-----< YSTU:LR_6_2 >-----
Building LR_6_2 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.1.0:exec (default-cli) @ LR_6_2 ---
Список L:
I O F Z D Y A E L X
В списке нет одинаковых элементов.

BUILD SUCCESS

Total time:  0.324 s
Finished at: 2023-04-07T07:08:20+03:00
-----

```

6.3

Создать программу, использующую в качестве поля структуру с полями типа boolean и int, которая формирует очередь и выводит ее на экран. Организовать поиск заданного элемента.

```

package ystu.lr_6_3;
class Struct{
    boolean bool;
    int date;
    Struct next;
    // Конструктор:
    Struct(boolean value_bool, int value_date){
        bool = value_bool;
        date = value_date;
    }
}

```

```

class List{
Struct pbeg = null;
Struct pend = null;

// Добавление первого элемента:
void first(boolean bool, int date){
    Struct elem = new Struct(bool, date);
    elem.next = null;
    pbeg = elem;
    pend = pbeg;
}

// Добавление в конец очереди:
void add(boolean bool, int date){
    Struct elem = new Struct(bool, date);
    elem.next = null;
    pend.next = elem;
    pend = elem;
}

// Удаление из начала очереди:
void pop (){
    pbeg=pbeg.next;
}

// Вывод на экран:
void print(){
    Struct elem = pbeg;
    while (elem != pend){
        System.out.print(elem.bool + " " + elem.date + "; ");
        elem = elem.next;
    }
    System.out.println();
}

// Поиск заданного элемента:
void seek(){
    int i = 1;
    int j = 0;
    Struct elem = pbeg;
    while (elem != pend){
        if (elem.date == 3){
            System.out.println("заданный элемент в очереди есть, его позиция: "
                + (i) + ", его значения: " + elem.bool + " " + elem.date);
            j++;
        }
    }
}

```



```

        break;
    }
    elem = elem.next;
    i++;
}
if (j == 0)
    System.out.println("заданного элемента в очереди нет.");
}
}
public class LR_6_3 {
    public static void main(String[] args) {
        List och = new List();
        och.first(false, 0);

        for (int i=1;i<10;i++){
            boolean bool;
            int date;
            int j = (int)(Math.random()*10%2);
            if (j == 0)
                bool = false;
            else bool = true;
            j = (int)(Math.random()*10);
            och.add(bool, j);
        }
        System.out.println("Очередь: ");
        och.print();
        System.out.println("\nОчередь после удаления первого элемента: ");
        och.pop();
        och.print();
        System.out.println("\nПоиск элемента с числовым значением 3, ближайшего
к началу очереди:");
        och.seek();
    }
}

```

```
Output - Run (LR_6_3) ×
--- exec-maven-plugin:3.1.0:exec (default-cli) @ LR_6_3 ---
Очередь:
false 0; true 2; true 4; true 5; true 3; true 3; false 8; true 0; false 7;

Очередь после удаления первого элемента:
true 2; true 4; true 5; true 3; true 3; false 8; true 0; false 7;

Поиск элемента с числовым значением 3, ближайшего к началу очереди:
заданный элемент в очереди есть, его позиция: 4, его значения: true 3
-----
BUILD SUCCESS
-----
Total time: 0.314 s
Finished at: 2023-04-14T16:14:15+03:00
-----
```

6.4

Создать программу, использующую в качестве поля структуру с полями типа `integer` и `float`, которая формирует стек и выводит его на экран. Организовать поиск заданного элемента.

```
package ystu.lr_6_4;
class Struct{
    int date_1;
    float date_2;
    Struct next;
    // Конструктор:
    Struct(int value_date_1, float value_date_2){
        date_1 = value_date_1;
        date_2 = value_date_2;
    }
}

class List{
    Struct top = null;

    // Добавление первого элемента:
    void first(int date_1, float date_2){
        Struct elem = new Struct(date_1, date_2);
        elem.next = null;
        top = elem;
    }

    // Занесение в стек:
    void push(int date_1, float date_2){
```

```

    Struct elem = new Struct(date_1, date_2);
    elem.next = top;
    top = elem;
}

```

// Удаление вершины стека:

```

void pop (){
    top = top.next;
}

```

// Вывод на экран:

```

void print(){
    Struct elem = top;
    while (elem != null){
        System.out.print(elem.date_1 + " и " + elem.date_2 + "; ");
        elem = elem.next;
    }
    System.out.println();
}

```

// Поиск заданного элемента:

```

void seek(){
    int i = 1;
    int j = 0;
    Struct elem = top;
    while (elem != null){
        if (elem.date_1 == 3){
            System.out.println("заданный элемент в стеке есть, его позиция: "
                + (i) + ", его значения: " + elem.date_1 + " и " + elem.date_2);
            j++;
            break;
        }
        elem = elem.next;
        i++;
    }
    if (j == 0)
        System.out.println("заданного элемента в стеке нет.");
}
}

```

```

public class LR_6_4 {
    public static void main(String[] args) {
        List stek = new List();
        stek.first(0, 0);

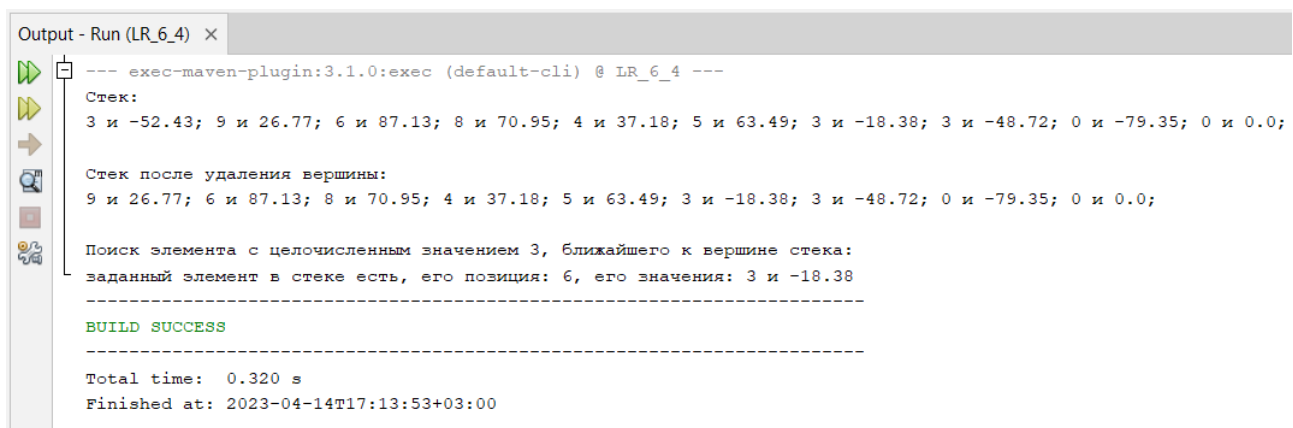
        for (int i=1;i<10;i++){

```

```

        int date_1;
        float date_2;
        date_1 = (int)(Math.random()*10);
        date_2 = (float)(200*Math.random()-100);
        date_2 = (float)(Math.round(date_2*100.0)/100.0);
        stek.push(date_1, date_2);
    }
    System.out.println("Стек: ");
    stek.print();
    System.out.println("\nСтек после удаления вершины: ");
    stek.pop();
    stek.print();
    System.out.println("\nПоиск элемента с целочисленным значением 3,
ближайшего к вершине стека:");
    stek.seek();
}
}

```



```

Output - Run (LR_6_4) x
--- ехес-maven-plugin:3.1.0:ехес (default-cli) @ LR_6_4 ---
Стек:
3 и -52.43; 9 и 26.77; 6 и 87.13; 8 и 70.95; 4 и 37.18; 5 и 63.49; 3 и -18.38; 3 и -48.72; 0 и -79.35; 0 и 0.0;

Стек после удаления вершины:
9 и 26.77; 6 и 87.13; 8 и 70.95; 4 и 37.18; 5 и 63.49; 3 и -18.38; 3 и -48.72; 0 и -79.35; 0 и 0.0;

Поиск элемента с целочисленным значением 3, ближайшего к вершине стека:
заданный элемент в стеке есть, его позиция: 6, его значения: 3 и -18.38
-----
BUILD SUCCESS
-----
Total time: 0.320 s
Finished at: 2023-04-14T17:13:53+03:00
-----

```

6.5

Создать двоичное дерево, элементы для которого взять из заданного целочисленного одномерного массива. В созданном двоичном дереве организовать поиск заданного элемента. В созданном двоичном дереве организовать просмотр его элементов любым способом.

```

package ystu.lr_6_5;
class Item{
    int d;
    Item left;
    Item right;
    Item(int value){
        d = value;
    }
}
class Tree{

```

```

Item root; // Корень дерева
// Формирование первого элемента дерева
void first(int value){
    Item pv = new Item(value);
    pv.left = null;
    pv.right = null;
    root = pv;
}
// Поиск с включением
void search_insert(int value){
    Item pv = root;
    Item prev = null;
    while(pv != null){
        prev = pv;
        if (value == -pv.d) return;
        else if (value < pv.d) pv = pv.left;
        else pv = pv.right;
    }
    // Создание нового узла
    Item pnew = new Item(value);
    pnew.left = null;
    pnew.right = null;
    if (value < prev.d)
        prev.left = pnew; // Присоединение к левому поддереву предка
    else
        prev.right = pnew; // Присоединение к правому поддереву предка
}
// Обход дерева
void print_tree (Item p){
    if (p != null){
        System.out.print(p.d + " ");
        print_tree (p.left);
        print_tree (p.right);
    }
}

// Поиск заданного элемента:
void seek (int value){
    Item pv = root;
    int i = 0;
    while(pv != null){
        if (value == pv.d){
            System.out.println("\nЗаданный элемент (" + value + ") в двоичном дереве
есть.");

```

```

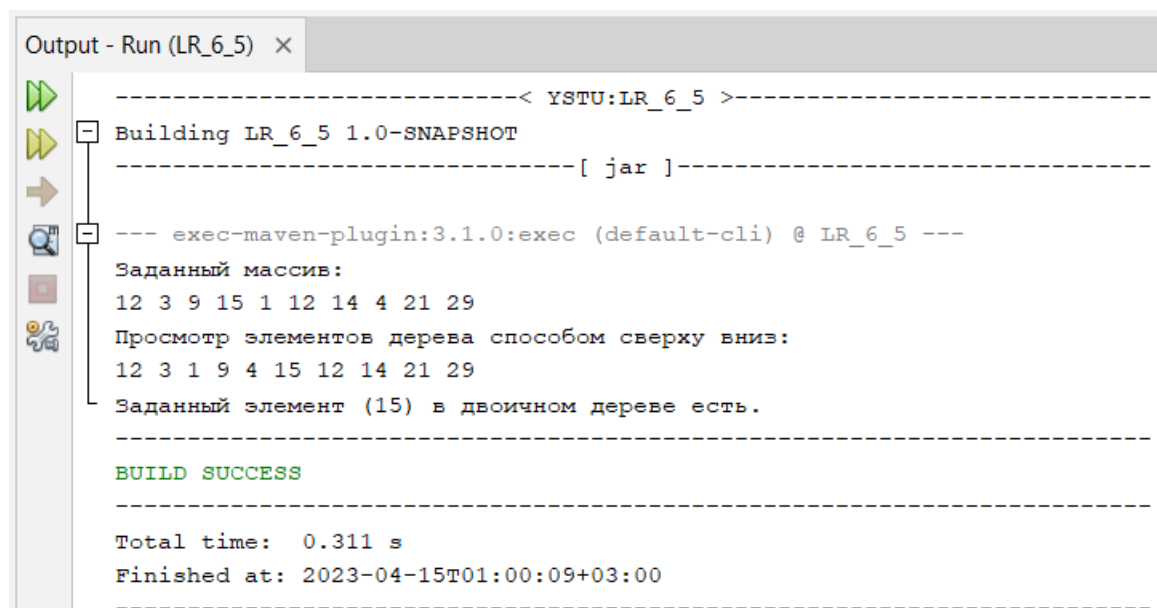
        i++;
        break;
    }
    else if (value < pv.d) pv = pv.left;
    else pv = pv.right;
}
if (i == 0)
    System.out.println("\nЗаданного элемента (" + value + ") в двоичном дереве
нет.");
}
}

```

```

public class LR_6_5 {
    public static void main(String[] args) {
        System.out.println("Заданный массив:");
        int a[] = new int[10];
        for (int i = 0; i < 10; i++){
            a[i] = (int)(Math.random()*30 + 1);
            System.out.print(a[i] + " ");
        }
        System.out.println();
        Tree m = new Tree();
        m.first(a[0]);
        for (int i = 1; i < 10; i++){
            m.search_insert(a[i]);
        }
        System.out.println("Просмотр элементов дерева способом сверху вниз:");
        m.print_tree(m.root);
        m.seek(15);
    }
}

```



```

Output - Run (LR_6_5) x
-----< YSTU:LR_6_5 >-----
Building LR_6_5 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.1.0:exec (default-cli) @ LR_6_5 ---
Заданный массив:
12 3 9 15 1 12 14 4 21 29
Просмотр элементов дерева способом сверху вниз:
12 3 1 9 4 15 12 14 21 29
Заданный элемент (15) в двоичном дереве есть.

BUILD SUCCESS

Total time: 0.311 s
Finished at: 2023-04-15T01:00:09+03:00
-----

```

Лабораторная работа №7

Создать суперкласс Транспортное средство и подклассы Автомобиль, Велосипед, Повозка. Подсчитать время и стоимость перевозки пассажиров и грузов каждым транспортным средством.

```
abstract class Transport{
    double time;
    double cost;
    String park [];
    void setParam (double timeTransport, double costTransport, String parkTransport
[]){
        time = timeTransport;
        cost = costTransport;
        park = parkTransport;
    }
    abstract String Park();
}
```

```
class Car extends Transport{
    @Override
    String Park(){
        return "автомобиль";
    }
    int speed = 80;
    int price = 40;
    Car (){
        setParam(0,0,null);
    }
    Car (int distanceCar, double ratioCar){
        double timeCar = (double)distanceCar/speed;
        double costCar = (double)distanceCar*ratioCar*price;
        String [] parkCar = new String [9];
        for (int i=0;i<9;i++)
            parkCar [i] = "A 00"+(i+1)+" BT";
        setParam(timeCar, costCar, parkCar);
    }
}
```

```
class Bicycle extends Transport{
    @Override
    String Park(){
        return "велосипед";
    }
    int speed = 30;
```

```

int price = 15;
Bicycle () {
    setParam(0,0,null);
}
Bicycle (int distanceBicycle, double ratioBicycle){
    double timeBicycle = (double)distanceBicycle/speed;
    double costBicycle = (double)distanceBicycle*ratioBicycle*price;
    String [] parkBicycle = new String [9];
    for (int i=0;i<9;i++)
        parkBicycle [i] = "B 00"+(i+1)+" ЕЛ";
    setParam(timeBicycle, costBicycle, parkBicycle);
}
}

```

```

class Wagon extends Transport{
    @Override
    String Park(){
        return "повозка";
    }
    int speed = 20;
    int price = 30;
    Wagon () {
        setParam(0,0,null);
    }
    Wagon (int distanceWagon, double ratioWagon){
        double timeWagon = (double)distanceWagon/speed;
        double costWagon = (double)distanceWagon*ratioWagon*price;
        String [] parkWagon = new String [9];
        for (int i=0;i<9;i++)
            parkWagon [i] = "П 00"+(i+1)+" ОБ";
        setParam(timeWagon, costWagon, parkWagon);
    }
}

```

```

class Park {
    String park(Transport T){
        return "К Вам придет "+T.Park()+" номер ";
    }
}

```

```

public class LR_7 extends javax.swing.JFrame {
    public LR_7() {
        initComponents();
    }
}

```



```
private void выходMenuItemActionPerformed(java.awt.event.ActionEvent evt) {  
    System.exit(0);  
}
```

```
private void oПрограммеMenuItemActionPerformed(java.awt.event.ActionEvent  
evt) {  
    JOptionPane.showMessageDialog(null, "<html><p  
align=center>Лабораторная работа 7"  
        + "<p align=center>Выполнил студент группы ДЦИС-27" + "<p  
align=center>Конторин П.Л.");  
}
```

```
private void расчетButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    int RadioButton = 0;  
    if (автомобильRadioButton.isSelected()) RadioButton = 1;  
    if (велосипедRadioButton.isSelected()) RadioButton = 2;  
    if (повозкаRadioButton.isSelected()) RadioButton = 3;  
    int distance = 0;  
    try{  
        distance = Integer.parseInt(jTextField1.getText());  
    }  
    catch(NumberFormatException e){  
        JOptionPane.showMessageDialog(null,  
            "<html><p align=center>Введите расстояние.");  
    }  
    double ratio = 1;  
    if (пассажирыCheckBox.isSelected())  
        ratio*=1.3;  
    if (грузыCheckBox.isSelected())  
        ratio*=1.2;  
    if (!пассажирыCheckBox.isSelected()&&!грузыCheckBox.isSelected()){  
        JOptionPane.showMessageDialog(null,  
            "<html><p align=center>Выберите пассажиры и/или грузы.");  
        Car car = new Car();  
        времяLabel.setText(""+car.time);  
        стоимостьLabel.setText(""+car.cost);  
        return;  
    }  
    int i = (int)(Math.random()*9);  
    switch (RadioButton){  
        case 1:  
            Car car = new Car(distance, ratio);  
            времяLabel.setText(""+Math.round(car.time*100.0)/100.0);  
            стоимостьLabel.setText(""+Math.round(car.cost));
```

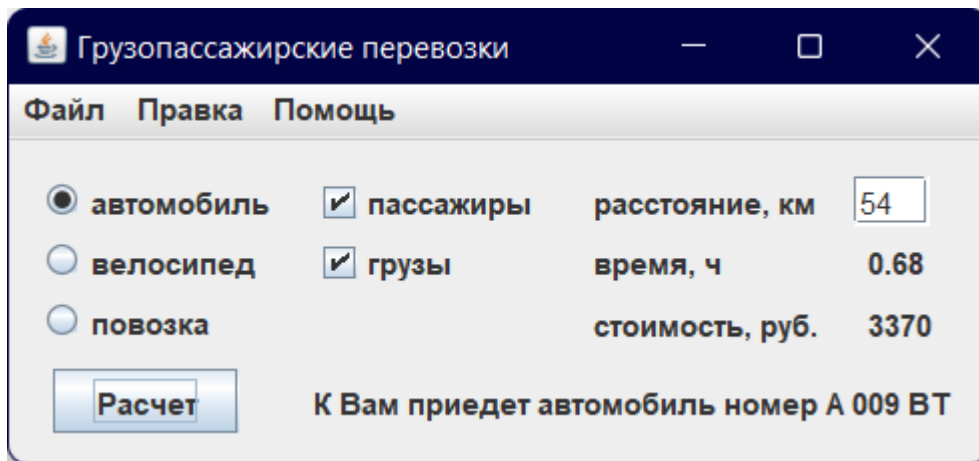
```

        String a1 = new String();
        Park b1 = new Park();
        a1 = b1.park(car);
        jLabel4.setText(a1 + car.park[i]);
        break;
    case 2:
        Bicycle bicycle = new Bicycle(distance, ratio);
        времяLabel.setText(""+Math.round(bicycle.time*100.0)/100.0);
        стоимостьLabel.setText(""+Math.round(bicycle.cost));
        String a2 = new String();
        Park b2 = new Park();
        a2 = b2.park(bicycle);
        jLabel4.setText(a2 + bicycle.park[i]);
        break;
    case 3:
        Wagon wagon = new Wagon(distance, ratio);
        времяLabel.setText(""+Math.round(wagon.time*100.0)/100.0);
        стоимостьLabel.setText(""+Math.round(wagon.cost));
        String a3 = new String();
        Park b3 = new Park();
        a3 = b3.park(wagon);
        jLabel4.setText(a3 + wagon.park[i]);
        break;
    }
}

private void очиститьMenuItemActionPerformed(java.awt.event.ActionEvent evt)
{
    Car car = new Car();
    времяLabel.setText(""+car.time);
    стоимостьLabel.setText(""+car.cost);
    jLabel4.setText("");
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            //new LR_7().setVisible(true);
            LR_7 frame = new LR_7();
            frame.setVisible(true);
            frame.setLocationRelativeTo(null);
        }
    });
}
}

```



Лабораторная работа №8

1. Изучить работу с пакетами
2. Создать приложение, в котором:
 - a. продемонстрировать умение работать с пакетами (доступ к именам из других пакетов, импорт пакетов);
 - b. продемонстрировать умение обрабатывать исключительные ситуации:
 - ✓ с использованием множественного блока catch();
 - ✓ с использованием вложенных блоков try();
 - ✓ использованием искусственного генерирования исключений;
 - ✓ с использованием выбрасывания исключений методами;
 - ✓ с использованием создания собственных исключений.

```
package package_1;
```

```
import javax.swing.*;
```

```
import package_3.package_4.*;
```

```
public class LR_8 extends javax.swing.JFrame {  
    public static int q;  
    public LR_8() {  
        initComponents();  
    }  
}
```

```
    public static void Message(int i){  
        switch(i){  
            case 0:  
                JOptionPane.showMessageDialog(null, "Индекс массива за пределами  
допустимых границ");  
                break;  
            case 1:  
                JOptionPane.showMessageDialog(null, "Присваивание элементу
```

```

массива недопустимого значения");
        break;
    case 2:
        JOptionPane.showMessageDialog(null, "Недопустимое приведение
типов");
        break;
    case 3:
        JOptionPane.showMessageDialog(null, "Недопустимое
использование нулевого указателя");
        break;
    case 4:
        JOptionPane.showMessageDialog(null, "Создание массива
отрицательного размера");
        break;
    case 5:
        JOptionPane.showMessageDialog(null, "Недопустимое
преобразование текстовой строки в числовой формат");
        break;
    case 6:
        JOptionPane.showMessageDialog(null, "Пароль верный!");
        break;
    }
}

private void выходMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}

private void oПрограммеMenuItemActionPerformed(java.awt.event.ActionEvent
evt) {
    JOptionPane.showMessageDialog(null, "<html><p align=center>Лабораторная
работа 8"
        + "<p align=center>Выполнил студент группы ДЦИС-27" + "<p
align=center>Конторин П.Л.");
}

private void
ArithmeticExceptionButtonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        package_2.Program p = new package_2.Program();
        int x = p.a/Program_4.b; // Арифметическая ошибка
    }
    catch(ArithmeticException e){
        JOptionPane.showMessageDialog(null, "Арифметическая ошибка");
    }
}

```

```
}
```

```
private void
```

```
ArrayIndexOutOfBoundsExceptionButtonActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    package_3.Program p = new package_3.Program();  
    p.multipleCatch(0);
```

```
}
```

```
private void
```

```
ArrayStoreExceptionButtonActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    package_3.Program p = new package_3.Program();  
    p.multipleCatch(1);
```

```
}
```

```
private void
```

```
ClassCastExceptionButtonActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    package_3.Program p = new package_3.Program();  
    p.multipleCatch(2);
```

```
}
```

```
private void
```

```
IllegalArgumentExceptionButtonActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    int age;  
    String ageStr = JOptionPane.showInputDialog("Введите возраст:");  
    int ageNum = Integer.parseInt(ageStr);  
    try {  
        IllegalArgumentException iaExc = new IllegalArgumentException();  
        if (ageNum <= 0) {  
            throw iaExc;  
        }  
    }  
    else {  
        age = ageNum;  
    }  
}
```

```
    catch(IllegalArgumentException e){  
        JOptionPane.showMessageDialog(null, "Методу передан недопустимый  
аргумент");  
    }  
}
```

```
}
```

```
private void
```

```
NullPointerExceptionButtonActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    package_2.Program p = new package_2.Program();
```

```

    p.i = 1;
    p.nestedTry();
}

```

```

private void
NegativeArraySizeExceptionButtonActionPerformed(java.awt.event.ActionEvent
evt) {
    package_2.Program p = new package_2.Program();
    p.nestedTry();
}

```

```

private void
NumberFormatExceptionButtonActionPerformed(java.awt.event.ActionEvent evt) {
    package_2.Program p = new package_2.Program();
    p.i = 2;
    p.nestedTry();
}

```

```

private void
IllegalAccessExceptionHandlerActionPerformed(java.awt.event.ActionEvent evt) {
    package_3.package_4.Program_4 p = new package_3.package_4.Program_4();

    try{
        p.throwOne();
    }
    catch(IllegalAccessException e){
        JOptionPane.showMessageDialog(null, "В доступе к классу
отказано");
    }
}

```

```

private void MyExceptionHandlerActionPerformed(java.awt.event.ActionEvent evt)
{
    String pass = JOptionPane.showInputDialog("Введите пароль (0000):");
    try{
        MyExceptionCheck.Pass(pass);
    }
    catch(MyException e){
        JOptionPane.showMessageDialog(null, "" + e);
    }
}

```

```

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {

```

```

        LR_8 frame = new LR_8();
        frame.setVisible(true);
        frame.setLocationRelativeTo(null);
    }
});
}
}

```

```
package package_2;
```

```
import package_1.LR_8;
```

```

public class Program{
    public int a = 14;
    public static int i = 0;
    public static void nestedTry(){
        try{
            int a[] = null;
            int b;
            int d;
            if (i == 1)
                b = a[2]; // Недопустимое использование нулевого указателя
            try{
                if (i == 2)
                    // Недопустимое преобразование текстовой строки в числовой формат:
                    d = Integer.parseInt("surprise");
                int c[] = new int[-1]; // Создание массива отрицательного размера
            }
            catch(NumberFormatException e){
                i = 0;
                LR_8.Message(5);
            }
        }
        catch(NullPointerException e){
            LR_8.Message(3);
            i = 0;
        }
        catch(NegativeArraySizeException e){
            LR_8.Message(4);
        }
    }
}
}

```

```
package package_3;
```

```
import package_1.*;
```

```

public class Program{
    public static void multipleCatch(int i){
        try{
            switch(i){
                case 0:
                    char[] ch = new char[2];
                    char ch1 = ch[2]; // Индекс массива за пределами допустимых границ
                    break;
                case 1:
                    Object x[] = new String[2];
                    x[0] = new int[0]; // Присваивание элементу массива недопустимого
значения
                    break;
                case 2:
                    Object obj = new java.util.Date();
                    Integer j = (Integer) obj; // Недопустимое приведение типов
                    break;
            }
        }
        catch(ArrayIndexOutOfBoundsException e){
            LR_8.Message(0);
        }
        catch(ArrayStoreException e){
            LR_8.Message(1);
        }
        catch(ClassCastException e){
            LR_8.Message(2);
        }
    }
}

```

```

package package_3.package_4;

```

```

public class MyException extends Exception{
    private String badPass;
    public MyException(String str){
        badPass=str;
    }
    @Override
    public String toString() {
        return "Введенный пароль " + badPass + " не верный!";
    }
}

```

```

package package_3.package_4;

```

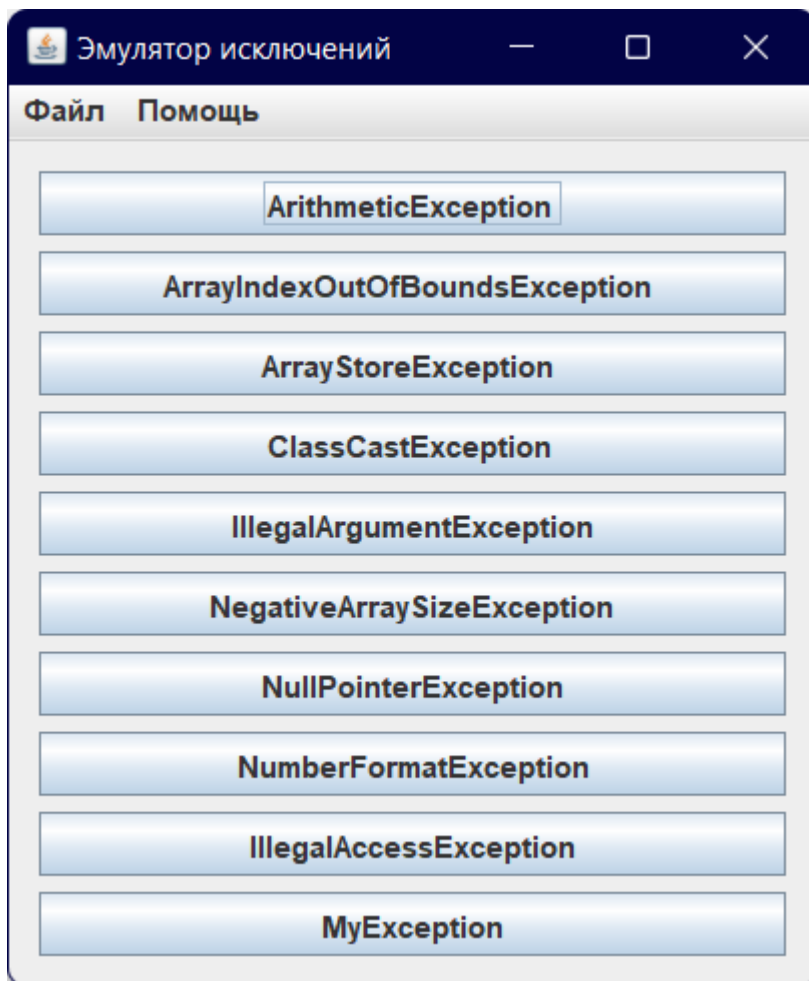


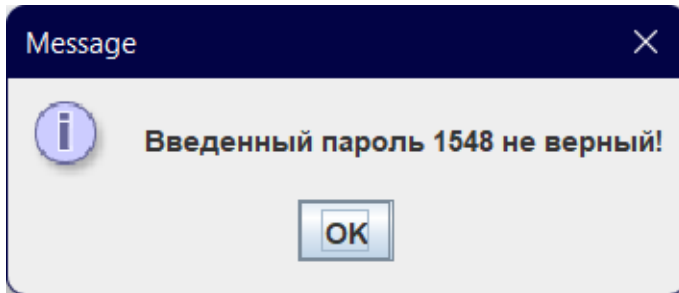
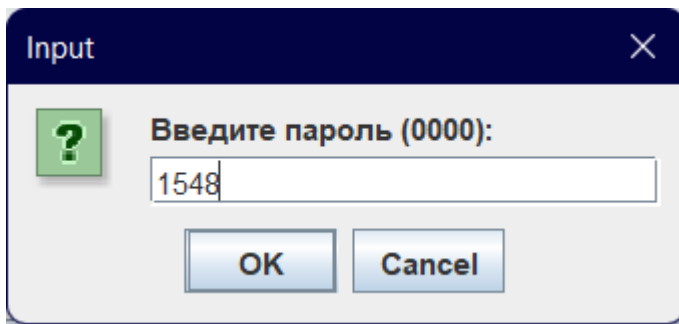
```
import package_1.LR_8;
```

```
public class MyExceptionCheck{  
    public static void Pass(String x) throws MyException{  
        if (x.equals("0000")) LR_8.Message(6);  
        else throw new MyException(x);  
    }  
}
```

```
package package_3.package_4;
```

```
public class Program_4{  
    public static int b = 0;  
    public static void throwOne() throws IllegalAccessException{  
        throw new IllegalAccessException();  
    }  
}
```





Лабораторная работа №9

9.1

Продemonстрировать создание нескольких потоков (thread) с синхронизацией. В одном потоке запустить задачу «вычисление бесконечного ряда», а в другом – задачу «табулирование функции».

```
package ystu.lr_9_1;
class MyPrint{
    synchronized void printValue(double pV_1, double pV_2, int time){
        try{
            Thread.sleep(time);
            System.out.print(pV_1 + "\t");
            Thread.sleep(2*time);
            System.out.println(pV_2);
        }
        catch(InterruptedException e){
            System.out.println("Прерывание потока: "+e);}
    }
}

class TabFunc implements Runnable{
    Thread t;
    MyPrint myPrint;
    int time;
    double a;
    double b;
    TabFunc(double a1, double b1, int time, MyPrint obj){
```

```

    a = a1;
    b = b1;
    myPrint = obj;
    this.time = time;
    t = new Thread(this);
    t.start();
}
public void run(){
    double x, y = 0;
    for (x=0; x<=7; x+=0.5)
    {
        if (x<2.3) y=(a+b)/(Math.exp(x)+Math.cos(x));
        else if (x<5) y=(a+b)*(x+1);
        else y=Math.exp(x)+Math.sin(x);
        myPrint.printValue(x, y, time);
    }
}
}

```

```

class SumInfRow implements Runnable{
    Thread t;
    MyPrint myPrint;
    int time;
    double x, S, z, T, E;
    int k;
    SumInfRow(double E1, int time, MyPrint obj){
        E = E1;
        myPrint = obj;
        this.time = time;
        t = new Thread(this);
        t.start();
    }
    public void run(){
        for (x=-Math.PI/4; x<=Math.PI/4; x+=0.05*Math.PI){
            T=4*x/(Math.PI+Math.PI*x*x);
            k=1;
            S=0;
            while (Math.abs(T)>E){
                S+=T;
                k++;
                T=4*x/(Math.PI*Math.pow((2*k-1),2)+Math.PI*x*x);
            }
            myPrint.printValue(x, S, time);
        }
    }
}

```

```
}
```

```
class LR_9_1 {  
    public static void main(String args[]) {  
        MyPrint obj = new MyPrint();  
        TabFunc tabFunc = new TabFunc(2.7, -0.27, 35, obj);  
        SumInfRow sumInfRow = new SumInfRow(0.001, 250, obj);  
        try {  
            tabFunc.t.join();  
            sumInfRow.t.join();  
        }  
        catch (InterruptedException e) {  
            System.out.println("Прерывание потока: "+e);  
        }  
    }  
}
```

```
Output - Run (LR_9_1) ×
-----< YSTU:LR_9_1 >-----
Building LR_9_1 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.1.0:exec (default-cli) @ LR_9_1 ---
0.0      1.215
0.5      0.9618795525114057
1.0      0.7457226512586987
1.5      0.5337812970070043
2.0      0.3484915561871757
2.5      8.505
3.0      9.72
3.5      10.935
4.0      12.15
4.5      13.365
5.0      147.45423482791347
5.5      243.98639193865
6.0      403.14937799453617
6.5      665.3567530324497
7.0      1097.2901450271772
-0.7853981633974483      -0.8280325758070912
-0.6283185307179586      -0.7417860021789444
-0.47123889803846897      -0.6167870779679284
-0.3141592653589793      -0.44698339161734146
-0.15707963267948966      -0.23472138161430955
0.0      0.0
0.15707963267948966      0.23472138161430955
0.3141592653589793      0.44698339161734146
0.47123889803846897      0.6167870779679284
0.6283185307179586      0.7417860021789444
0.7853981633974483      0.8280325758070912

BUILD SUCCESS

Total time:  10.429 s
Finished at: 2023-04-25T09:30:01+03:00
-----
```

9.2

Продемонстрировать ввод с клавиатуры в консольном режиме. Для выполнения варианта задания на одномерные массивы в консольном режиме исходный массив ввести с клавиатуры.

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

```
public class LR_9_2 {
```

```
    public static void main(String[] args) throws IOException {
        double cmin, sumao=0;
        int i, j, imin=0, isumao=0;
```

```

// Ввод с клавиатуры массива a:
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
System.out.println("Введите через пробел целочисленные значения массива
a:");
String a_str = br.readLine();
String[] a_str_mas = a_str.split(" ");
int a[] = new int [a_str_mas.length+1];
for (i=0;i<a_str_mas.length;i++)
    a[i] = Integer.parseInt(a_str_mas [i]);
/* Поиск заданного значения методом последовательного перебора
с использованием барьерного элемента:*/
// Ввод с клавиатуры заданного для поиска значения:
System.out.print("Введите искомое число: ");
int x = Integer.parseInt(br.readLine());
for(i=0;a[a.length-1]=x;a[i]!=x;i++){
    if (i==a.length-1) System.out.println("Искомое число "+x+" в массиве
нет.");
    else System.out.println("Искомое число "+x+" в массиве есть, его индекс:
"+i);
// Сортировка массива a выбором наименьшего элемента:
i=0;
while (i<a.length-2){
    int min=i;
    j=i+1;
    while (j<a.length-1){
        if (a[j]<a[min]) min=j;
        j++;
    }
    int buf=a[i];
    a[i]=a[min];
    a[min]=buf;
    i++;
}
System.out.println("Отсортированный массив a:");
for (i=0;i<a.length-1;i++){
    System.out.print(a[i]+" ");
}
System.out.println("\n");

// Ввод с клавиатуры массива b:
Scanner in = new Scanner(System.in);
System.out.println("Введите через пробел целочисленные значения массива
b:");
String b_str = in.nextLine();
String[] b_str_mas = b_str.split(" ");

```

```

int b[] = new int [b_str_mas.length];
for (i=0;i<b_str_mas.length;i++)
    b[i] = Integer.parseInt(b_str_mas [i]);
// Сортировка массива b методом вставки:
for (i=1;i<b.length;i++){
    int buf=b[i];
    for (j=i;j>0&&buf<b[j-1];j--){
        b[j]=b[j-1];
    }
    b[j]=buf;
}
System.out.println("Отсортированный массив b:");
for (i=0;i<b.length;i++)
    System.out.print(b[i]+" ");
System.out.println();
// Бинарный поиск:
// Ввод с клавиатуры заданного для поиска значения:
System.out.print("Введите искомое число: ");
x = Integer.parseInt(in.nextLine());
int left=0;
int right=b.length-1;
int middle=(left+right)/2;
while (left<=right&&b[middle]!=x){
    if (x>b[middle]) left=middle+1;
    else right = middle - 1;
    middle=(left+right)/2;
}
if (left>right) System.out.println("Искомое число "+x+" в массиве нет.\n");
else System.out.println("Искомое число "+x+" в массиве есть, его индекс:
"+middle+"\n");

// Слияние массивов a и b
System.out.println("Слияние массивов a и b:");
int ab[]=new int [a.length-1+b.length];
int ia, ib, iab;
for (ia=ib=iab=0;iab<a.length-1+b.length;iab++){
    if ( ia < a.length-1 ){
        if ( ib < b.length ){
            if (a[ia]<b[ib])
                ab[iab]=a[ia++];
            else ab[iab]=b[ib++];
        }
        else ab[iab]=a[ia++];
    }
    else ab[iab]=b[ib++];
}

```

```

    }
    for (i=0;i<a.length-1+b.length;i++)
        System.out.print(ab[i]+" ");
    System.out.println("\n");

    // Ввод с клавиатуры массива с:
    System.out.println("Введите через пробел значения массива с в виде чисел с
плавающей запятой:");
    String c_str = in.nextLine();
    String[] c_str_mas = c_str.split(" ");
    double c[] = new double [c_str_mas.length];
    for (i=0;i<c_str_mas.length;i++)
        c[i] = Double.parseDouble(c_str_mas [i]);
    // Поиск индекса минимального по модулю элемента:
    cmin=c[0];
    for(i=0;i<c.length;i++){
        if (Math.abs(c[i])<Math.abs(cmin)){
            imin=i;
            cmin=c[i];
        }
    }
    System.out.println("Номер минимального по модулю элемента массива с:
"+(imin+1));
    // Поиск индекса элемента, расположенного после первого отрицательного
элемента:
    for(i=0;i<c.length;i++){
        if (c[i]<0) {
            isumao=i+1;
            break;
        }
    }
    // Подсчет суммы модулей элементов, расположенных после первого
отрицательного элемента:
    if (isumao==0) System.out.println("В массиве с отрицательных элементов
нет.");
    else{
        for(i=isumao;i<c.length;i++) sumao+=Math.abs(c[i]);
        System.out.print("Сумма модулей элементов массива с, расположенных
после первого отрицательного элемента: ");
        System.out.println(String.format("%.2f", sumao));
    }
}
}

```



```
Output x
Run (LR_9_2) x Run (LR_9_2) x
-----< YSTU:LR_9_2 >-----
Building LR_9_2 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.1.0:exec (default-cli) @ LR_9_2 ---
Введите через пробел целочисленные значения массива a:
45 346 -519 13 84 -754 88 330 -72 11
Введите искомое число: 13
Искомое число 13 в массиве есть, его индекс: 3
Отсортированный массив a:
-754 -519 -72 11 13 45 84 88 330 346

Введите через пробел целочисленные значения массива b:
-75 840 72 -64 666 14 -70 71 -8
Отсортированный массив b:
-75 -70 -64 -8 14 71 72 666 840
Введите искомое число: 2
Искомого числа 2 в массиве нет.

Слияние массивов a и b:
-754 -519 -75 -72 -70 -64 -8 11 13 14 45 71 72 84 88 330 346 666 840

Введите через пробел значения массива c в виде чисел с плавающей запятой:
4.8e2 0.81e-3 7.13e4 -5.11e-3 3.15 -7.15e-1 4.01 8.684e3
Номер минимального по модулю элемента массива c: 2
Сумма модулей элементов массива c, расположенных после первого отрицательного элемента: 8691,88
-----
BUILD SUCCESS
-----
Total time: 03:26 min
Finished at: 2023-04-26T06:24:55+03:00
-----
```

9.3

Продemonстрировать работу с файлами (чтение и запись) с использованием графического диалогового режима. Создать окно с меню. В меню должны быть пункт «Файл». В пункте «Файл» должны быть подпункты «Открыть», «Сохранить как...», «Выход».

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

```
public class LR_9_3 extends javax.swing.JFrame {
    public LR_9_3() {
        initComponents();
    }
}
```

```
public class SimpleFileFilter extends javax.swing.filechooser.FileFilter {
    String ext;
    SimpleFileFilter(String ext){
        this.ext=ext;
    }
    public boolean accept(File f){
```

```

if(f==null)
    return false;
if(f.isDirectory()){
    return true ;
}
else
    return (f.getName().endsWith(ext));
}
public String getDescription(){
return "Text files (.txt)";
}
}

```

File file;

```

javax.swing.JFileChooser fileChooser = new javax.swing.JFileChooser();
javax.swing.filechooser.FileFilter fileFilter = new SimpleFileFilter(".txt");

```

```

private void выходMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}

```

```

private void оПрограммеMenuItemActionPerformed(java.awt.event.ActionEvent
evt) {
    JOptionPane.showMessageDialog(null, "<html><p align=center>Лабораторная
работа 9.3"
        + "<p align=center>Выполнил студент группы ДЦИС-27" + "<p
align=center>Конторин П.Л.");
}

```

```

private void открытьMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    File folder = new File("D:/Java");
    fileChooser.setCurrentDirectory(folder);
    fileChooser.addChoosableFileFilter(fileFilter);
    if(fileChooser.showOpenDialog(null)!= fileChooser.APPROVE_OPTION)
        return;
    file = fileChooser.getSelectedFile();
    try{
        InputStream fileInpStream = new FileInputStream(file);
        int size = fileInpStream.available();
        fileInpStream.close();
        char[] buff = new char[size];
        Reader fileReadStream = new FileReader(file);
        fileReadStream.read(buff);
        JTextArea1.setText(String.valueOf(buff));
        fileReadStream.close();
    }
}

```

```

    } catch(Exception e){
        javax.swing.JOptionPane.showMessageDialog(null,
            "Ошибка чтения из файла \n"+file.getAbsolutePath());
    }
}

private void сохранитьКакMenuItemActionPerformed(java.awt.event.ActionEvent
evt) {
    fileChooser.addChoosableFileFilter(fileFilter);
    if(fileChooser.showSaveDialog(null)!=fileChooser.APPROVE_OPTION){
        return;
    };
    file = fileChooser.getSelectedFile();
    try{
        Writer filWriteStream=new FileWriter(file);
        filWriteStream.write(jTextArea1.getText() );
        filWriteStream.close();
    } catch(Exception e){
        javax.swing.JOptionPane.showMessageDialog(null,
            "Ошибка записи в файл \n"+file.getAbsolutePath());
    }
}

private void очиститьMenuItemActionPerformed(java.awt.event.ActionEvent evt)
{
    jTextArea1.setText("");
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            LR_9_3 frame = new LR_9_3();
            frame.setVisible(true);
            frame.setLocationRelativeTo(null);
        }
    });
}
}

```

