

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ  
Кафедра веб-технологий и компьютерного моделирования**

---

**А. С. Кравчук, А. И. Кравчук, Е. В. Кремень**

**Введение  
в язык JAVA.  
Сборник заданий  
и тематических примеров**

**Учебные материалы  
для студентов специальности 1-31 03 08  
«Математика и информационные технологии  
(по направлениям)»**

---

**МИНСК  
2023**

УДК 004.432.045:004.738.5Java (075.8)

ББК 32.973.2-018.1я73-1

К78

Рекомендовано советом  
механико-математического факультета БГУ  
26 января 2023 г., протокол № 5

Рецензент  
кандидат технических наук, доцент *М. Н. Садовская*

**Кравчук, А. С.**

К78 Введение в язык Java. Сборник заданий и тематических примеров : учеб. материалы для студентов спец. 1-31 03 08 «Математика и информационные технологии (по направлениям)» / А. С. Кравчук, А. И. Кравчук, Е. В. Кремень. – Минск : БГУ, 2023. – 54 с.

Содержатся задачи по программированию начиная от создания первой программы до циклов и массивов. В каждой теме приводятся примеры решения типовых задач и варианты индивидуальные заданий. Учебные материалы предназначены для отработки основных конструкций языка Java, правил алгоритмизации. Издание ориентировано в первую очередь на тех, кто не имеет опыта практического программирования на языке Java, адресуется студентам, а также всем, кто хотел бы научиться приемам программирования стандартных задач.

УДК 004.432.045:004.738.5Java (075.8)

ББК 32.973.2-018.1я73-1

© Кравчук А. С., Кравчук А. И.,  
Кремень Е. В., 2023  
© БГУ, 2023

## Оглавление

Введение.....	5
Тема 1. Онлайн интегрированные среды. Операции. Операторы управления программой.....	6
Часть 1. Запуск онлайн интегрированной среды разработки.....	6
Часть 2. Создание первой программы.....	6
Часть 3. Изучение операций в Java.....	7
Часть 4. Программирование линейных и разветвляющихся алгоритмов.....	9
Задания для программирования линейных алгоритмов.....	10
Задания для программирования разветвляющиеся алгоритмы.....	11
Часть 5. Оператор цикла <code>for()</code> .....	11
Часть 6. Оператор цикла <code>while()</code> . Знакомство с классом <code>Math</code> .....	13
Тема 2. Обработка одномерных массивов.....	15
Тема 3. Применение методов для решения задач с разветвляющимися алгоритмами.....	17
Часть 1. Использование экземплярного метода.....	17
Часть 2. Использование статического метода.....	20
Тема 4. Перечисления <code>enum</code> . Оператор выбора <code>switch()</code> .....	23
Часть 1. Общее знакомство с <code>enum</code> и <code>switch()</code> .....	23
Часть 2. Методы использующие <code>enum</code> и <code>switch()</code> .....	26
Тема 5. Использование операторов циклов в методах.....	29
Часть 1. Вычисление в методах конечных сумм и произведений.....	29
Часть 2. Цепочка вызовов методов. Табулирование значений математической функции.....	32
Тема 6. Создание класса-библиотеки методов для циклических вычислений с известным числом повторений.....	35
Часть 1. Создание экземплярного метода класса-библиотеки.....	37
Часть 2. Создание статического метода класса-библиотеки.....	38

Тема 7. Разработка класса, содержащего методы реализующие итерационные вычисления.....	39
Часть 1. Вычисление с помощью статического метода бесконечной суммы ряда с заданной точностью .....	39
Часть 2. Определение предела сходящейся последовательности в экземплярном методе.....	41
Тема 8. Создание класса, содержащего методы, реализующие рекурсию.....	43
Часть 1. Статические методы, реализующие рекурсию .....	43
Часть 2. Экземплярные методы, реализующие рекурсию.....	45
Тема 9. Создание класса, содержащего методы обработки строк.....	46
Часть 1. Задачи определения числовых характеристик заданной строки .....	49
Часть 2. Удаление (забой) символа в строке сдвигом влево .....	50
Часть 3. Циклическая перестановка символов в подстроке.....	51
Часть 4. Зеркальная (реверс) перестановка символов в подстроке.....	52
Литература.....	54

## Введение

Современные тенденции все более широкого использования информационных технологий и средств компьютерного моделирования во всех отраслях экономики ставят новые требования по уровню подготовленности студентов высших учебных заведений в области программирования.

Создание у студентов достаточной теоретической и практической базы по этим предметам позволит им быстро включаться в решение актуальных задач, стоящих перед предприятиями, занятыми разработкой программного обеспечения.

Поскольку данная отрасль экономики является не только престижной для Республики Беларусь, но и демонстрирует хорошую динамику развития, то перед высшими учебными заведениями стоит задача обеспечить ее как можно более квалифицированными кадрами, а также всемерно увеличить выпуск специалистов данного направления, хорошо владеющих мульти-парадигмальным подходом, основанным на использовании прежде всего принципов объектно-ориентированного и функционального программирования.

При этом, в связи со всеобщей современной тенденцией создания кроссплатформенных приложений, одним из наиболее востребованных языков является Java. Цель данной публикации является систематизация этапов получения студентами практических навыков по разработке программ с использованием этого языка.

Предлагаемый список тем, примеров и заданий к ним соответствует первичному ознакомлению студентов с языком Java.

Материалы издания могут быть использованы как при проведении занятий в аудитории, так и с использованием системы электронного обучения и тестирования Moodle. Кроме того, оценка возможности решения заданий из всего пула задач по теме может помочь осуществлять проверку результативности усвоения материала определенной темы студентами самостоятельно.

# Тема 1. Онлайн интегрированные среды. Операции. Операторы управления программой

## Часть 1. Запуск онлайн интегрированной среды разработки.

Можно воспользоваться одной из двух ссылок:

- JDoodle Online Java Compiler IDE (<https://www.jdoodle.com/online-java-compiler/>);
- OnlineGDB ([https://www.onlinegdb.com/online\\_java\\_compiler](https://www.onlinegdb.com/online_java_compiler)).

Отличительные особенности интегрированных сред:

- среда JDoodle – более предпочтительна, т.к. позволяет разрабатывать консольные приложения даже для многофайловых проектов с использованием пакетов, но не позволяет организовать наглядную работу с файлами.
- среда OnlineGDB позволяет продемонстрировать практически все особенности разработки, кроме демонстрации работы с пакетами в многофайловых проектах.

### Замечание.

*Среда OnlineGDB требует, чтобы класс, содержащий программу в виде метода `main()` всегда имел название `Main`, среда JDoodle в базовой реализации этого не требует.*

Как и в случае работы с онлайн интегрированными средами, если программа не прошла компиляцию, то в нижней части каждой из сред располагается окно со списком ошибок.

## Часть 2. Создание первой программы

Для первичного ознакомления особенностями использования онлайн компиляторов и создания первой программой на Java следует набрать следующий код.

### Пример программы.

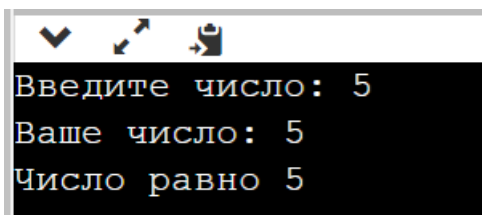
```
Main.java
1 import java.util.Scanner;
2
```

```

3 public class Main {
4     public static void main(String[] args) {
5         //сообщение
6         System.out.print("Введите число: ");
7         //организация ввода числа
8         Scanner objIn = new Scanner(System.in);
9         int num = objIn.nextInt();
10        //сообщение
11        System.out.printf("Ваше число: %d \n", num);
12        //еще один вариант вывода результата
13        System.out.println("Число равно " + num);
14        objIn.close();
15    }
16 }

```

Результат работы программы, если с клавиатуры ввести число 5:



```

Введите число: 5
Ваше число: 5
Число равно 5

```

### Часть 3. Изучение операций в Java.

Рассмотрим пример выполнения задания.

#### Пример программы.

```

Main.java
1 import java.util.Scanner;
2
3 public class Main {
4     public static void main(String[] args) {
5         Scanner objIn = new Scanner(System.in);
6         System.out.print("Введите строку: ");
7         String str = objIn.nextLine();
8         System.out.println("Строка: " + str);
9         float x, y, z;
10        System.out.print("Введите x, y, z: ");
11        x = objIn.nextFloat();

```

```

12     y = objIn.nextFloat();
13     z = objIn.nextFloat();
14     int k,m;
15     System.out.print("Введите целые k, m: ");
16     k = objIn.nextInt();
17     m = objIn.nextInt();
18     objIn.close();
19     boolean a, b, c, d;
20     a = m % 3 == 1;
21     b = 2 != x * z * y;
22     c = k / 2 >= 5;
23     d = true;
24     System.out.println("a = " + a +
25                         "b = " + b +
26                         "\n\tc = " + c +
27                         "\n\td = " + b);
28     System.out.println("Результат выражения: " +
29                         (!((a || b) && c) == ((! b) && d)));
30 }
31 }

```

### Результат работы программы:

```

Введите строку: Добрый день!
Строка: Добрый день!
Введите x, y, z: 1 2 3
Введите целые k, m: 4 5
a = false b = true
      c = false
      d = true
Результат выражения: false

```

### Общее задание:

1. Определить результат определения остатка от целочисленного деления  $a \% k$ , где  $a = 100$ , а  $n$  – номер студента в списке подгруппы.
2. Определить результат битового сдвига *вправо* (переменные целочисленные) в выражении  $a \gg k$ , где  $a = 256$ , а  $k = n \% 5$  ( $n$  – номер студента в списке подгруппы).



3. Определить результат битового сдвига *влево* (переменные целочисленные) в выражении  $a \gg k$ , где  $a = 2$ , а  $k = n \% 5$  ( $n$  – номер студента в списке подгруппы)
4. Вычислить значения выражений:
  - 4.1.  $((6 \% 3 == 1) == (0 == 1));$
  - 4.2.  $((2 != 2*3*4) \&\& (2 != 24));$
  - 4.3.  $(( (1 \& 3) | 2) * 8);$
  - 4.4.  $((1 == 2) \& (2 == 8 \% 3));$
  - 4.5.  $((5 / 2 >= 5) || (2 >= 5));$
  - 4.6.  $((8 \% 3 == 1) == (2 == 1));$
  - 4.7.  $((2 != 1*1*2) == (!false != true));$
  - 4.8.  $('d'-'c') * 2.5;$

## Часть 4. Программирование линейных и разветвляющихся алгоритмов

**Пример задания.** Определить, попадает ли точка внутрь круга, если радиус круга (*radius*) известен, а его центр (*xCircle*, *yCircle*) и точка (*xPoint*, *yPoint*) задаются своими координатами.

### Пример программы.

Main.java

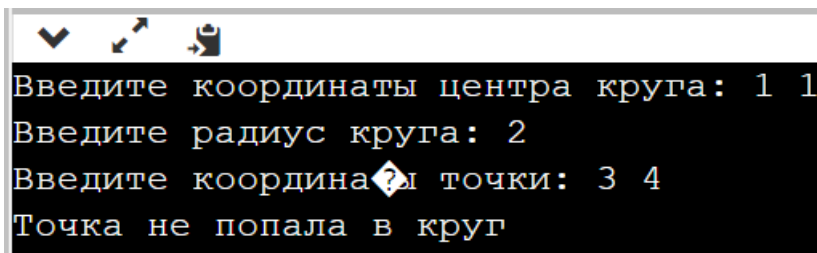
```
1 import java.util.Scanner;
2
3 public class Main {
4     public static void main(String[] args) {
5         Scanner objIn = new Scanner(System.in);
6         //координаты центра круга
7         float xCircle, yCircle;
8         System.out.print("Введите координаты центра " +
9             "         "         "круга: ");
10        xCircle = objIn.nextFloat();
11        yCircle = objIn.nextFloat();
12        //радиус круга
13        float radius;
14        System.out.print("Введите радиус круга: ");
15        radius = objIn.nextFloat();
16        //координаты точки
17        float xPoint, yPoint;
```

```

18     System.out.print("Введите координаты точки: ");
19     xPoint = objIn.nextFloat();
20     yPoint = objIn.nextFloat();
21     objIn.close();
22     if ( Math.pow(xPoint - xCircle, 2) +
23         Math.pow(yPoint - yCircle, 2) <=
24         Math.pow(radius,2) ) {
25         System.out.println("Точка принадлежит кругу");
26     }
27     else {
28         System.out.println("Точка не попала в круг");
29     }
30 }
31 }

```

### Результат работы программы:



```

Введите координаты центра круга: 1 1
Введите радиус круга: 2
Введите координаты точки: 3 4
Точка не попала в круг

```

## Задания для программирования линейных алгоритмов

### Варианты индивидуальных заданий.

1. Даны  $x, y$ . Получить  $\frac{|x|-|y|}{1+|xy|}$ .
2. Даны два числа. Найти среднее арифметическое кубов этих чисел и среднее геометрическое модулей этих чисел.
3. Вычислить расстояние между двумя точками с данными координатами.
4. Найти площадь треугольника со сторонами  $a, b, c$  по формуле Герона:

$$S = \sqrt{p(p-a)(p-b)(p-c)}, \text{ где } p = \frac{(a+b+c)}{2}.$$

5. Даны  $x, y, z$ . Найти  $w$ , если:  $w = \sin \left( \left( y - \sqrt{|x|} \right) \left( x - \frac{y}{z^2 + \frac{x^2}{4}} \right) \right)$

6. Даны  $x, y, z$ . Найти  $w$ , если:  $w = x - \frac{x^2}{1 + \sin^2(x + y + z)}$
7. Даны  $x, y, z$ . Найти  $w$ , если:  $w = \cos\left(z^2 + \frac{x^2}{4} + y\right)$
8. Даны  $x, y, z$ . Найти  $w$ , если:  $w = y + x - \frac{\cos^2 x}{1 + \sqrt{|y + z|}}$

### Задания для программирования разветвляющиеся алгоритмы

#### Варианты индивидуальных заданий.

1. Даны два числа. Вывести первое число, если оно больше второго, и оба числа, если это не так.
2. Найти наименьшее из трех данных чисел.
3. Даны три числа, являющиеся длинами сторон треугольника. Определить тип треугольника (равносторонний, равнобедренный, разносторонний).
4. Даны  $x, y, z$ . Найти:  $\max\{x + y + z, xyz\} + 3$ .
5. Даны  $x, y, z$ . Найти:  $\min\{x^2 + y^2, y^2 + z^2\} - 4$ .
6. Дано  $x$ . Вычислить  $y$ , если:  $y = \begin{cases} x^2, & \text{при } -2 \leq x \leq 2, \\ 4, & \text{при } x < -2 \text{ и } x > 2 \end{cases}$
7. Дано  $x$ . Вычислить  $y$ , если:  $y = \begin{cases} 0, & \text{при } x \leq 0 \\ x, & \text{при } 0 < x \leq 1, \\ x^4, & \text{при } x > 1 \end{cases}$
8. Дано  $x$ . Вычислить  $y$ , если:  $y = \begin{cases} x^2 + 4x + 5, & \text{при } x \leq 2, \\ \frac{1}{x^2 + 4x + 5}, & \text{при } x > 2 \end{cases}$
9. Дано  $x$ . Вычислить  $y$ , если:  $y = \begin{cases} 0, & \text{при } x \leq 0 \\ x^2 - x, & \text{при } 0 < x \leq 1, \\ x^2 - \sin \pi x^2 - 1, & \text{при } x > 1 \end{cases}$

### Часть 5. Оператор цикла for ()

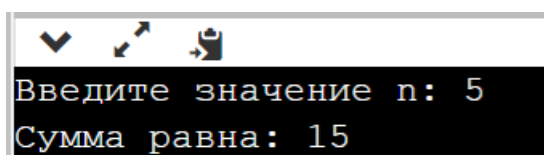
**Пример задания.** Даны натуральное  $n$ . Вычислить значение суммы  $s$  по формуле  $s = 1 + 2 + \dots + n$ .

## Пример программы.

Main.java

```
1 import java.util.Scanner;
2
3 public class Main {
4     public static void main(String[] args) {
5         Scanner objIn = new Scanner(System.in);
6         int n, sum;
7
8         while (true) {
9             System.out.print("Введите значение n: ");
10            n = objIn.nextInt();
11            if (n > 0) break;
12            System.out.print("Введено отрицательное n!!!"+
13                "\nПопробуйте еще раз!!!");
14        }
15        objIn.close();
16        sum = 0;
17        for(int i = 1; i <= n; i++) {
18            sum = sum + i;
19        }
20        System.out.print("Сумма равна: " + sum);
21    }
22 }
```

## Результат работы программы:



```
Введите значение n: 5
Сумма равна: 15
```

## Варианты индивидуальных заданий.

Дано натуральное  $n$ . Вычислить значение суммы или произведения:

- 1)  $\frac{1}{1} + \frac{1}{2} + \dots + \frac{1}{n}$ ;
- 2)  $2^n$ ;
- 3)  $\left(1 + \frac{1}{1^2}\right) \left(1 + \frac{1}{2^2}\right) \dots \left(1 + \frac{1}{n^2}\right)$ ;
- 4)  $\frac{1}{1 \cdot 2} - \frac{1}{2 \cdot 3} + \dots + \frac{(-1)^{n+1}}{n(n+1)}$ ;

- 5)  $\frac{1}{1^5} + \frac{1}{2^5} + \dots + \frac{1}{n^5}$ ;
- 6)  $\frac{1}{3^2} + \frac{1}{5^2} + \dots + \frac{1}{(2n+1)^2}$ ;
- 7)  $-\frac{1}{3} + \frac{1}{5} + \dots + \frac{(-1)^n}{2n+1}$ ;
- 8)  $\frac{1}{\sin 1} + \frac{1}{\sin 1 + \sin 2} + \dots + \frac{1}{\sin 1 + \sin 2 + \dots + \sin n}$ .

## Часть 6. Оператор цикла `while()`. Знакомство с классом `Math`

**Обобщенная формулировка задания.** Вычислить  $m$  значений заданной математической функции  $f(x)$  на отрезке  $[a, b]$ . Результаты оформить в виде таблицы 1. Значения  $a, b, h$  ввести с клавиатуры.

Таблица 1

$x_i$	$f(x_i)$
$x_0 = a$	
...	
...	
$x_{m-1} = b$	

Столбцы таблицы:

- первый столбец – значение координаты  $x_i$ ;
- второй столбец – значения математической функции  $f(x_i)$ , вычисленное с использованием методов класса `Math`.

Замечание.

Для построения таблицы использовать символы '=', '-', '|' и т.д.

**Пример задания.** Построить таблицу значений функции  $e^x$  на отрезке  $[0,1]$  с шагом  $h$ .

**Пример программы.**

```
Main.java
1 import java.util.Scanner;
2
3 public class Main {
4     public static void main(String[] args) {
```

```

5 Scanner objIn = new Scanner(System.in);
6 double a, b, h, x;
7 String str = "\n\t===== ";
8
9 while (true) {
10     System.out.print("Введите значения a, b, h: ");
11     a = objIn.nextDouble();
12     b = objIn.nextDouble();
13     h = objIn.nextDouble();
14     if ((a < b) && (h > 0) && (h < b - a)) break;
15     System.out.print("Параметры введены неверно!" +
16                     "\nПопробуйте еще раз!");
17 }
18 //формирование шапки таблицы
19 System.out.printf("\tТаблица значений функции");
20 System.out.printf(str);
21 System.out.printf("\n\t|    x    | знач. функции |");
22 System.out.printf(str);
23
24 x = a;
25 while(x <= b) {
26     System.out.printf("\n\t| %5.3f | %7.5f |",
27                     x, Math.exp(x));
28     x = x + h;
29 }
30 System.out.printf(str);
31 }
32 }

```

### Результат работы программы:

```

Введите значения a, b, h: 1 2 0.2
Таблица значений функции
=====
|    x    | знач. функции |
=====
|  1.000  |    2.71828    |
|  1.200  |    3.32012    |
|  1.400  |    4.05520    |
|  1.600  |    4.95303    |
|  1.800  |    6.04965    |
|  2.000  |    7.38906    |
=====

```

### Варианты индивидуальных заданий.

№	$f(x)$	$[a, b]$	№	$f(x)$	$[a, b]$
1	$e^{-x}\sqrt{x}$	1;3	9	$\ln(x)$	1;1.5
2	$x\sqrt{x}$	3;4	10	$1 + \ln^2(x)$	0.4;1
3	$1/\sqrt{x}$	4;5	11	$1 + e^x$	0.5;1
4	$(2x+1)/\sqrt{x}$	5;7	12	$e^{x^2}/2$	2;3
5	$\sqrt[4]{x}$	6;8	13	$\cos(x)e^{-x}$	1;2
6	$\sqrt{x^2+1}$	8;9	14	$1/(1+e^{-x})$	3;4
7	$x \cdot \sin(x)$	0.5;1	15	$\sqrt[3]{x}$	1;5
8	$\cos(5 \cdot x)$	2.5;3	16	$x^{3.7}$	2;3

## Тема 2. Обработка одномерных массивов

**Пример задания.** Дан одномерный целочисленный массив. Найти и вывести на экран минимальный и максимальный элементы массива.

### Пример программы.

```
Main.java
1 import java.util.Scanner;
2
3 public class Main {
4     public static void main(String[] args) {
5         Scanner objIn = new Scanner(System.in);
6         //количество элементов массива
7         int n;
8         //константы, определяющие границы генерации значений
9         final int MAX = 25;
10        final int MIN = 5;
11        //проверка корректности ввода количества элементов
12        while (true) {
13            System.out.println("Введите количество элементов");
14            n = objIn.nextInt();
15            if ((n > 0)) break;
16            System.out.println("Количество элементов " +
17                "указано не верно!\n" +
18                "Попробуйте еще раз!");
19        }
20        objIn.close();
21        //создание и заполнение массива случ. числами
```

```

22     int[] a = new int[n];
23     for(int i = 0; i < n; i++) {
24         a[i] = (int) (Math.random() * (MAX - MIN) + MIN);
25     }
26     //вывод на экран исходного массива
27     for(int i = 0; i < n; i++) {
28         System.out.print(" " + a[i]);
29     }
30     //определение индексов максимального и минимального
31     //элементов массива
32     int maxInd = 0, minInd = 0;
33     for(int i = 0; i < a.length ; i ++ ) {
34         if ( a[i] < a[minInd] ) {
35             minInd = i;
36         }
37         else if ( a[i] > a[maxInd] ) {
38             maxInd = i;
39         }
40     }
41     System.out.print("\nМинимальный элемент = " +
42                     a[minInd] +
43                     "\nМаксимальный элемент = " +
44                     a[maxInd]);
45 }
46 }

```

### Результат работы программы:

```

Введите количество элементов
5
 22 9 22 11 22
Минимальный элемент = 9
Максимальный элемент = 22

```

### Варианты индивидуальных заданий.

#### Замечание

При выполнении индивидуального задания исходный массив заполнить случайными целыми числами. Исходный массив и полученный результат вывести на экран.

1. В одномерном массиве из  $n$  элементов найти порядковые номера первого отрицательного и последнего положительного элементов



- (если таковые имеются). Значение элементов и их порядковые номера вывести на экран или выдать соответствующее сообщение.
2. Ввести одномерный массив из  $n$  элементов. Вычислить сумму всех отрицательных чисел, их количество и сумму всех положительных чисел.
  3. В зависимости от того, образуют элементы заданного массива целых чисел из  $n$  элементов строго убывающую, не возрастающую, строго возрастающую, неубывающую последовательность, выдать соответствующее сообщение.
  4. Элемент называется локальным минимумом (максимумом), если у него нет соседа, меньшего (большего), чем он сам. Найти все локальные минимумы и максимумы в заданном массиве из  $n$  элементов.
  5. В последовательности из  $n$  элементов найти количество элементов, меньших заданного числа и вывести их на экран.
  6. Дан вектор размерности  $n$ . Пронормировать его по максимальному элементу (найти максимальный по модулю элемент и разделить все элементы массива на найденное значение).
  7. В массиве из  $n$  элементов посчитать количество элементов встречающихся только один раз.
  8. В массиве из  $n$  элементов посчитать количество нулевых элементов.
  9. В массиве из  $n$  элементов посчитать количество положительных элементов.
  10. В массиве из  $n$  элементов посчитать количество отрицательных элементов.

## Тема 3. Применение методов для решения задач с разветвляющимися алгоритмами

### Часть 1. Использование экземплярного метода

**Пример задания.** Определить, попадает ли точка в замкнутый круг заданного радиуса (глобальная константа), если центр круга лежит в начале координат (Рисунок 1), а точка задается своими координатами.

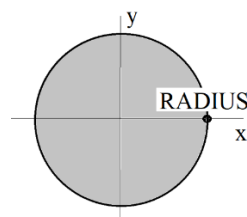
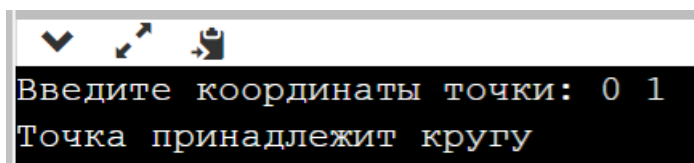


Рисунок 1 – Область, для которой проверяется попала ли точка

## Пример программы.

```
Main.java
1 import java.util.Scanner;
2
3 public class Main {
4
5     boolean BelongsCircle(float x, float y, double r) {
6         return x * x + y * y <= Math.pow(r, 2);
7     }
8
9     public static void main(String[] args) {
10        Scanner objIn = new Scanner(System.in);
11        //радиус круга
12        final double RADIUS = 2;
13        //координаты точки
14        float xPoint, yPoint;
15        System.out.print("Введите координаты точки: ");
16        xPoint = objIn.nextFloat();
17        yPoint = objIn.nextFloat();
18        objIn.close();
19        Main obj = new Main();
20        if ( obj.BelongsCircle(xPoint, yPoint, RADIUS) ) {
21            System.out.println("Точка принадлежит кругу");
22        }
23        else {
24            System.out.println("Точка не попала в круг");
25        }
26    }
27 }
```

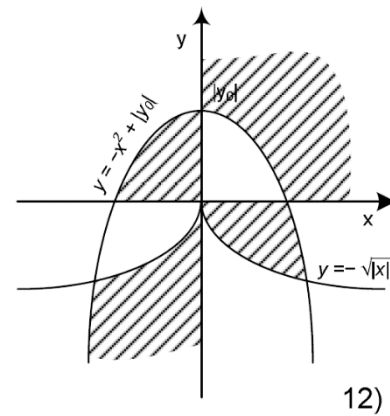
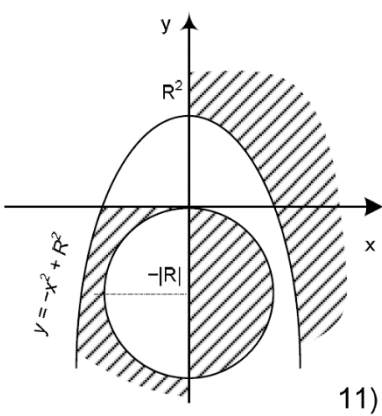
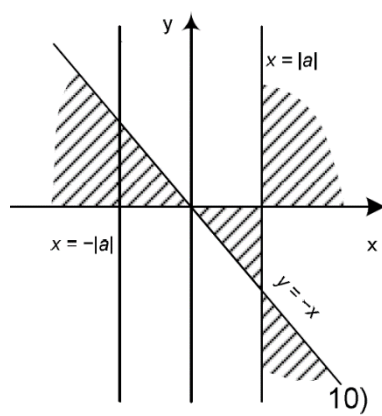
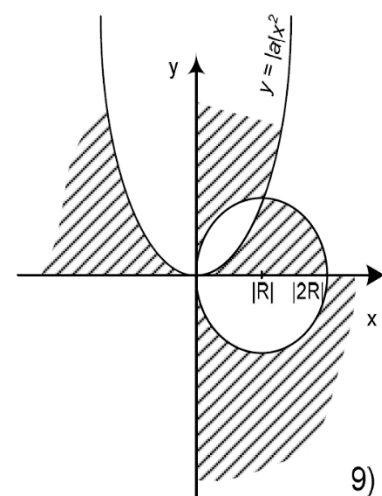
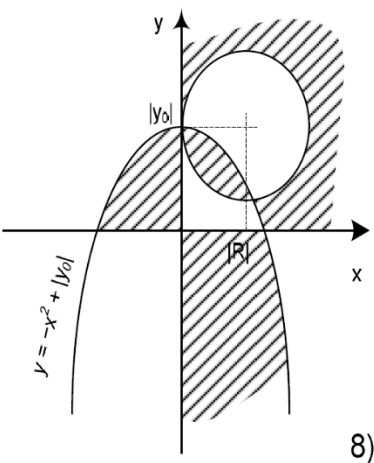
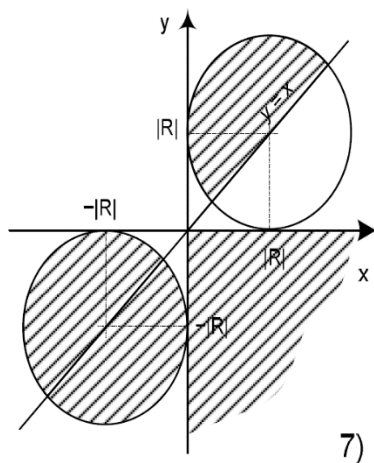
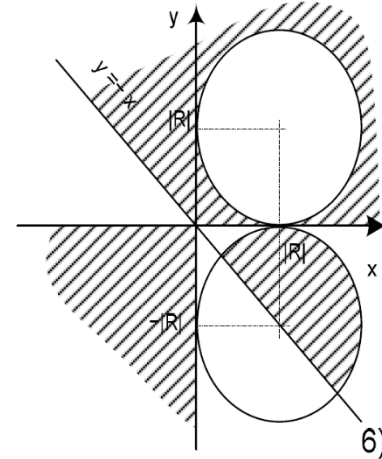
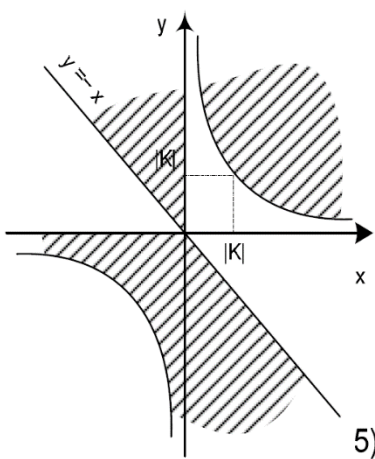
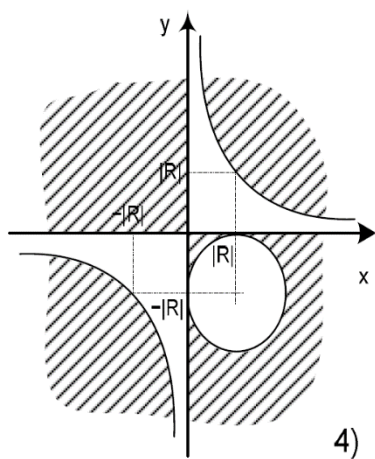
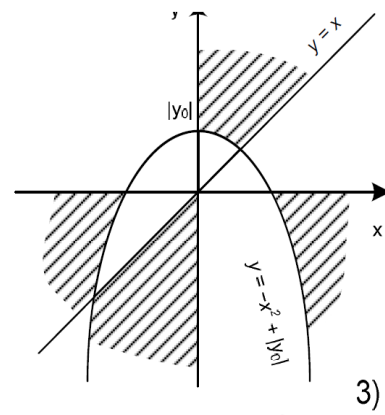
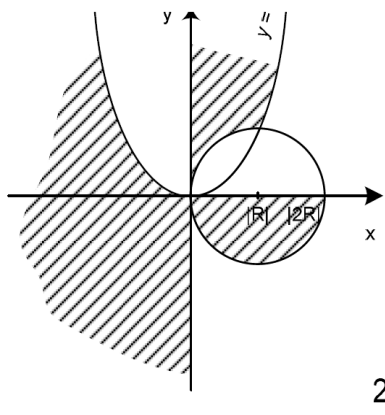
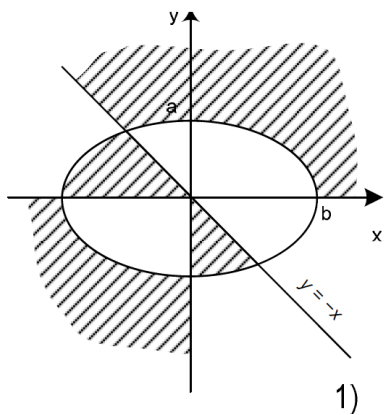
## Результат работы программы:

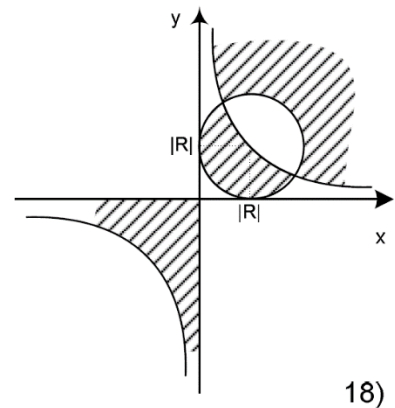
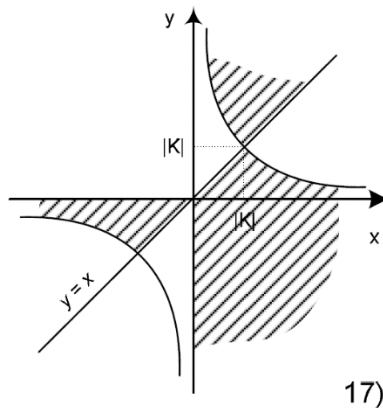
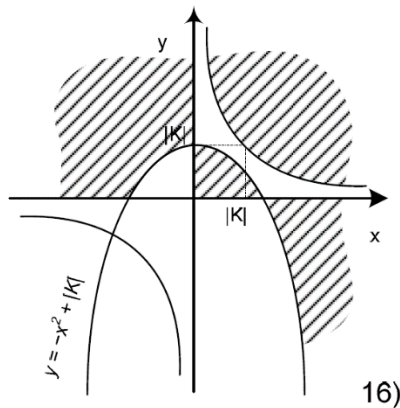
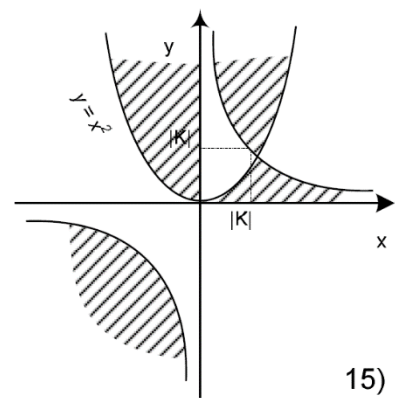
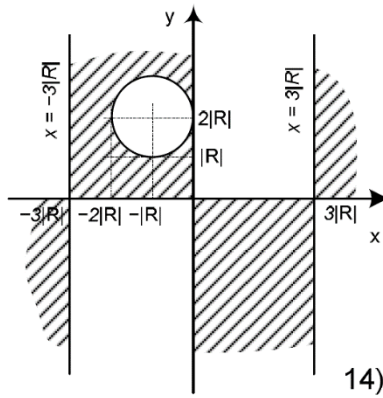
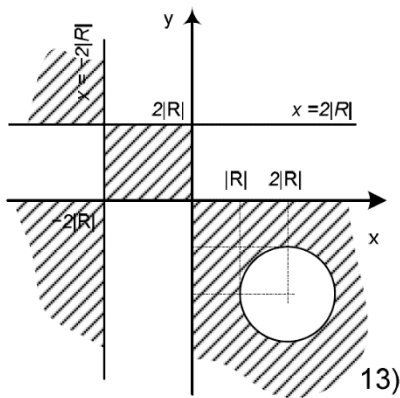


```
▼ ↗ 📄
Введите координаты точки: 0 1
Точка принадлежит кругу
```

## Варианты индивидуальных заданий.

Выдать соответствующее сообщение о принадлежности заданной точки заштрихованной области.





## Часть 2. Использование статического метода

**Пример задания.** Ввести число  $x$ , выяснить что больше: целая часть числа  $x$ , или его дробная часть, умноженная на 10. Если первое, то число разделить на 10, если второе, то дробную часть от него отнять.

### Пример программы.

Main.java

```

1 import java.util.Scanner;
2 import java.math.*; //для BigDecimal и MathContext
3
4 public class Main {
5
6     static double Operation(double x) {
7         double result;
8         //рассчитываем целую часть
9         int intPart = (int) x;;
10        //опред. дробную часть
11        double fractPart = x - (int)x;
12        //проверка условия
13        if ((double)intPart > fractPart *10) {
14            //для устранения ошибок операций с операндами с
15            //плавающей точкой переводим в BigDecimal с помощью

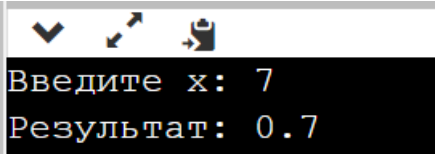
```

```

16         //MathContext() ограничиваем количество позиций в
17         //BigDecimal
18         BigDecimal xBd =
19             new BigDecimal(x, new MathContext(10));
20         //после деления возвращаем результат в double
21         result = xBd.divide(BigDecimal.TEN).doubleValue();
22     }
23     else {
24         result = x - fractPart;
25     }
26     return result;
27 }
28
29 public static void main(String[] args) {
30     Scanner objIn = new Scanner(System.in);
31     //радиус круга
32     double x;
33     System.out.print("Введите x: ");
34     x = objIn.nextDouble();
35     objIn.close();
36     System.out.println("Результат: " + Operation(x));
37 }
38 }

```

### Результат работы программы:



```

Введите x: 7
Результат: 0.7

```

### Варианты индивидуальных заданий.

1. Ввести три целые числа  $a, b, c$ . Вывести сумму чисел  $a + b + c$ , определить нечетная ли она, если нечетная, то найти значение произведения  $a \cdot b \cdot c$ .
2. Ввести целые числа  $a, b$ . Определить делится ли целое число  $a$  на целое число  $b$  без остатка. Если делится, вывести на экран их произведение  $a \cdot b$ .
3. Ввести целое число  $a$  получить число  $x$  равное:
  - половине  $a$ , если  $a$  – нечетное;
  - утроенному значению  $a$ , если  $a$  – четное.
4. Ввести целые числа  $a$  и  $b$  и действительное число  $z$ . Преобразовать число  $z$  по формуле  $z \cdot x$ , если  $a$  делиться на  $b$  без остатка и  $z/x$  в противном случае (где  $x$  – остаток от деления  $a$  на  $b$ ).

5. Ввести целое  $m$ . Если  $2^m$  больше, чем  $8 \cdot m$ , то  $m$  увеличить на 20, в противном случае  $m$  уменьшить в три раза.
6. Ввести неравные целые числа  $m, n$ . Определить какая из дробей  $m/n$  или  $n/m$  ближе к числу  $\pi$ . В первом случае число  $m$  умножить на 10, во втором удвоить число  $n$ . (использовать определенную пользователем константу `const double PI = 3.141592653589793;`)
7. Определить делится ли на 7 введенное целое число  $a$ . Если делится, то другому числу  $u$  присвоить 200, в противном случае  $u$  сделать равным 500.
8. Ввести целые числа  $m, n, k$ . Для двух дробей  $m/n$  и  $n/k$  выяснить равны ли их дробные части. Если дробные части равны, то вывести оба значения, если нет, то ограничиться выводом соответствующего сообщения.
9. Ввести целые числа  $m, n, l, k$ . Для двух дробей  $m/n$  и  $l/k$  выяснить равны ли их дробные части. Если части равны, то вывести один раз значение дробной части, если не равны, то вывести обе дробные части, умноженные на 2.
10. Ввести целые числа  $m, n$ . Для дроби  $m/n$  определить превышает ли дробная часть числа 0.7. Если превышает, то число  $m$  разделить на 2, иначе удвоить число  $n$ .
11. Ввести целые числа  $m, n$ . Определить превышает ли умноженная на 20 дробная часть числа  $m/n$  целую часть  $m/n$ . Если превышает, то вывести значение на сколько.
12. Ввести целые числа  $m, n$ . Определить является ли целая часть числа  $m/n$  нечетной. Если она нечетная, то  $m$  возвести в квадрат, иначе  $n$  утроить.
13. Ввести целые числа  $m, n$ . Возвести во вторую степень число  $m/n$ , если его целая часть больше числа  $k$ , где  $k$  остаток от деления  $m$  на 8.
14. Ввести три числа  $x, y, z$ , определить их сумму. Если сумма больше или равна 20, то оставить числа без изменения, в противном случае определить число  $s$  - сколько не хватает в сумме, чтобы она была равна 120.
15. Ввести число  $x$ . Выяснить превышает ли модуль числа  $\cos(x^2)$  значение  $\sqrt{2} / 2$ . Если превышает, то  $x$  уменьшить втрое, в противном случае  $x$  удвоить.



## Тема 4. Перечисления enum. Оператор выбора switch ()

### Часть 1. Общее знакомство с enum и switch ()

**Пример задания.** Дано натуральное (целое, беззнаковое) число rating. Вывести строку-описание оценки, соответствующей числу rating (1 - «плохо», 2..3 - «неудовлетворительно», 4..6 - «удовлетворительно», 7..8 - «хорошо», 9..10 - «отлично»). Если rating не лежит в диапазоне 1-10, то вывести строку «ошибка ввода». Использовать оператор выбора switch () и перечисления enum.

#### Пример программы.

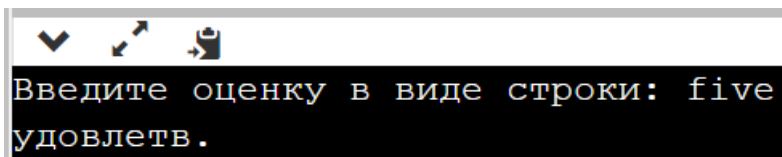
```
1 import java.util.Scanner;
2
3 enum Rating {
4     ONE,
5     TWO,
6     THREE,
7     FOUR,
8     FIVE,
9     SIX,
10    SEVEN,
11    EIGHT,
12    NINE,
13    TEN
14 }
15
16 public class Main
17 {
18     public static void main(String[] args) {
19         Scanner objIn = new Scanner(System.in);
20         System.out.print("Введите оценку в виде строки: ");
21         String str = objIn.nextLine();
22         objIn.close();
23         //метод toUpperCase() переводит строку в верхний рег.
24         //метод valueOf преобразует строку к типу перечисления
25         Rating grade = Rating.valueOf(str.toUpperCase());
26         switch(grade) {
27             case ONE: System.out.println("плохо");
28                 break;
29             case TWO:
30             case THREE: System.out.println("неудовлетв.");
31                 break;
32             case FOUR:
```

```

29     case TWO:
30     case THREE: System.out.println("неудовлетв.");
31                 break;
32     case FOUR:
33     case FIVE:
34     case SIX: System.out.println("удовлетв.");
35                 break;
36     case SEVEN:
37     case EIGHT: System.out.println("хорошо");
38                 break;
39     case NINE:
40     case TEN: System.out.println("отлично");
41                 break;
42     default:
43         System.out.println("Ошибка ввода");
44     }
45 }
46 }

```

### Результат выполнения программы:



### Варианты индивидуальных заданий.

#### Замечание.

Для выполнения задания необходимо либо использовать метод `ordinal()`, возвращающий индекс перечислителя в перечислении, либо воспользоваться примером к следующему заданию.

Перечислители в перечислении в Java «по умолчанию» имеют только индексы, а значений у них нет, пока не будет создан соответствующий конструктор для присвоения значений и методы работы с присвоенными значениями.

1. Дано целое число  $K$ . Вывести строку-описание оценки, соответствующей числу  $K$  (1 - «плохо», 2 - «неудовлетворительно», 3 - «удовлетворительно», 4 - «хорошо», 5 - «отлично»). Если  $K$  не лежит в диапазоне 1-5, то вывести строку «ошибка». Использовать оператор выбора `switch()` и перечисления `enum`.
2. Дан номер месяца - целое число в диапазоне 1-12 (1 - январь, 2 - февраль и т. д.). Вывести название соответствующего времени



- года («зима», «весна», «лето», «осень»). Использовать оператор выбора `switch()` и перечисления `enum`.
3. Ввести номер дня недели и вывести соответствующие ему название дня недели. Использовать оператор выбора `switch()` и перечисления `enum`.
  4. Ввести номер месяца и вывести соответствующее ему название месяца. Использовать оператор выбора `switch()` и перечисления `enum`.
  5. Введите время (только часы). Выведете соответствующее приветствие: «Доброе утро», «добрый день», «добрый вечер», «доброй ночи». Использовать оператор выбора `switch()` и перечисления `enum`.
  6. Дано число  $x$  с плавающей точкой. Выведете на экран меню типа (введите номер действия: 1 - возвести число в квадрат; 2 - извлечь корень квадратный; 3 - вычислить синус; 4 - косинус). И произведите эти действия. Использовать оператор выбора `switch()` и перечисления `enum`.
  7. Для натурального числа  $k$  вывести фразу «мы нашли  $k$  грибов в лесу», согласовав окончание слова «гриб» с числом  $k$ . Использовать оператор выбора `switch()` и перечисления `enum`.
  8. Дан номер месяца - целое число в диапазоне 1–12 (1 - январь, 2 - февраль и т. д.). Вывести на экран количество дней в этом месяце для не високосного года. Использовать оператор выбора `switch()` и перечисления `enum`.
  9. Вывести значение переменной  $k$  (от 0 до 10) римскими цифрами. Использовать оператор выбора `switch()` и перечисления `enum`.
  10. Напишите программу, выводящую сообщение о возможных сообщениях (1 - фамилия; 2 — имя; 3 — отчество; 4 — год рождения). Использовать оператор выбора `switch()` и перечисления `enum`.
  11. Ввести символ в нижнем регистре от  $a$  до  $f$ . Вывести на экран соответствующую букву в заглавном начертании. Выдать сообщение о неверном вводе, если введено любое другое значение, кроме оговоренных в задаче. Использовать оператор выбора `switch()` и перечисления `enum`.
  12. Вывести на экран вместо введенного целого числа от 0 до 9 соответствующее слово. Использовать оператор выбора `switch()` и перечисления `enum`.
  13. Ввести символ в верхнем регистре латинского алфавита от  $A$  до  $F$ . Вывести на экран соответствующую букву в строчном начертании. Выдать сообщение о неверном вводе, если введено любое другое

- значение, кроме оговоренных в задаче. Использовать оператор выбора `switch()` и перечисления `enum`.
14. Дан порядковый номер месяца, вывести на экран количество месяцев, оставшихся до конца года. Использовать оператор выбора `switch()` и перечисления `enum`.
  15. С 1 января этого года по некоторый день прошло  $n$  дней, определить название текущего месяца. Использовать оператор выбора `switch()` и перечисления `enum`.
  16. Дано число  $M$  - номер месяца. Определить номер квартала по введенному номеру месяца и номер полугодия. Использовать оператор выбора `switch()` и перечисления `enum`.

## Часть 2. Методы использующие `enum` и `switch()`

**Пример задания.** Даны два вещественных числа  $a$  и  $x$ . Определив принадлежность переменных  $a$  и  $x$  к определенным интервалам, используя экземплярный и статический методы (оба должны присутствовать в программе) вывести значение  $y$ , вычисленное по заданной формуле:

$$y = \begin{cases} \sin(a \cdot x), & -1 \leq a \leq 1 \text{ и } -1 \leq x \leq 1, \\ \cos(a \cdot x), & a < -1 \text{ и } 1 < a \text{ и } x < -1 \text{ и } 1 < x. \end{cases}$$

### Пример программы.

```

Main.java
1 import java.util.Scanner;
2
3 enum numberSet {
4     FIRST_SET,
5     SECOND_SET,
6     ERROR_SET
7 }
8
9 public class Main
10 {
11     //экземплярный метод
12     numberSet instanceDetectorSet(double a,
13                                 double x) {
14         numberSet result = numberSet.ERROR_SET;
15         if (-1 <= a && a <= 1 && -1 <= x && x <= 1 )
16             result = numberSet.FIRST_SET;
17         if ( a < -1 && 1 < a && x < -1 && 1 < x )
18             result = numberSet.SECOND_SET;
19         return result;
20     }

```

```

21 //статический метод
22 static numberSet staticDetectorSet(double a,
23                                     double x) {
24     numberSet result = numberSet.ERROR_SET;
25     if (-1 <= a && a <= 1 && -1 <= x && x <= 1 )
26         result = numberSet.FIRST_SET;
27     if ( a < -1 && 1 < a && x < -1 && 1 < x )
28         result = numberSet.SECOND_SET;
29     return result;
30 }
31
32 public static void main(String[] args) {
33     Scanner objIn = new Scanner(System.in);
34     System.out.print("Введите число с плав. точкой a ");
35     double a = objIn.nextDouble();
36     System.out.print("Введите x (число с плав. точкой) ");
37     double x = objIn.nextDouble();
38     objIn.close();
39     //применение экземплярного метода
40     Main obj = new Main();
41     System.out.print("Результат вызова экземп. метода ");
42     switch( obj.instanceDetectorSet(a, x) ) {
43         case FIRST_SET: System.out.println("sin(a*x) = " +
44                                     Math.sin(a * x));
45                         break;
46         case SECOND_SET: System.out.println("cos(a*x) = " +
47                                     Math.cos(a * x));
48                         break;
49         default: System.out.println("ошибка ввода");
50     }
51     //применение статического метода
52     System.out.print("Результат вызова статич. метода ");
53     switch( staticDetectorSet(a, x) ) {
54         case FIRST_SET: System.out.println("sin(a*x) = " +
55                                     Math.sin(a * x));
56                         break;
57         case SECOND_SET: System.out.println("cos(a*x) = " +
58                                     Math.cos(a * x));
59                         break;
60         default: System.out.println("ошибка ввода");
61     }
62 }
63 }

```

### Результаты работы программы:

```

input
Введите число с плав. точкой a 0.4
Введите x (число с плав. точкой) 0.7
Результат вызова экземп. метода sin(a*x) = 0.2763556485641137
Результат вызова статич. метода sin(a*x) = 0.2763556485641137

```

**Варианты индивидуальных заданий.**

N	Формула для вычислений
1.	$Y = \begin{cases} A \sin^2 X & \text{при } -5 \leq AX \leq 0 \text{ или } 3 \leq AX \leq 5; \\ A \cos X^2 & \text{при } 0 < AX < 3; \\ \ln(1 +  X A ) & \text{при } AX < -5 \text{ или } AX > 5, A \text{ и } X \text{ даны.} \end{cases}$
2.	$Y = \begin{cases} \sqrt{A \cos^2 X} & \text{при } -4 \leq AX \leq -2 \text{ или } 0 \leq AX \leq 4; \\ A^2 \cos X^2 + 1 & \text{при } -2 < AX < 0; \\ \ln(1 +  X + A^2 ) & \text{при } AX < -4 \text{ или } AX > 4, A \text{ и } X \text{ даны.} \end{cases}$
3.	$Y = \begin{cases} \sqrt{A + \operatorname{Tg} A^2 X} & \text{при } -10 \leq AX \leq -5 \text{ или } 0 \leq AX \leq 5; \\ 1 - A \cos X^3 & \text{при } -5 < AX < 0; \\ A + \ln^4(1 +  X ) & \text{при } AX < -10 \text{ или } AX > 5, A \text{ и } X \text{ даны.} \end{cases}$
4.	$Y = \begin{cases} \sqrt{A^2 + \operatorname{Ctg} AX} & \text{при } -10 \leq AX \leq -7 \text{ или } 0 \leq AX \leq 7; \\ 1 + A \sqrt{\cos X^3} & \text{при } -7 < AX < 0; \\ A + \operatorname{Tg}^4  XA  & \text{при } AX < -4 \text{ или } AX > 4, A \text{ и } X \text{ даны.} \end{cases}$
5.	$Y = \begin{cases} A + \sin^2 X & \text{при } -7 \leq AX \leq 0 \text{ или } 3 \leq AX \leq 7; \\ A \sqrt{\cos X^2} & \text{при } 0 < AX < 3; \\ \ln^4(1 +  X A ) & \text{при } AX < -4 \text{ или } AX > 4, A \text{ и } X \text{ даны.} \end{cases}$
6.	$Y = \begin{cases} \sqrt{1 + A \operatorname{Tg} X^2} & \text{при } -5 \leq AX \leq 0 \text{ или } 3 \leq AX \leq 5; \\ \ln  A + X  & \text{при } 0 < AX < 3; \\ A^2 + X^3 - 5 & \text{при } AX < -5 \text{ или } AX > 5, A \text{ и } X \text{ даны.} \end{cases}$
7.	$Y = \begin{cases} \sqrt{A^3 \cos X} & \text{при } -8 \leq AX \leq -3 \text{ или } 0 \leq AX \leq 4; \\ A^2 \cos X^2 + 1 & \text{при } -3 < AX < 0; \\ \ln  X + A^2  & \text{при } AX < -8 \text{ или } AX > 4, A \text{ и } X \text{ даны.} \end{cases}$
8.	$Y = \begin{cases} \sqrt{A + \sin A^2 X} & \text{при } -15 \leq AX \leq -5 \text{ или } 0 \leq AX \leq 15; \\ 1 + A \operatorname{Ctg} X^3 & \text{при } -5 < AX < 0; \\ A^3 + \ln(1 +  X ) & \text{при } AX < -15 \text{ или } AX > 15, A \text{ и } X \text{ даны.} \end{cases}$
9.	$Y = \begin{cases} \sqrt{A^3 + X} & \text{при } -6 \leq AX \leq -3 \text{ или } 0 \leq AX \leq 12; \\ A^2 \operatorname{Tg} X^2 + 1 & \text{при } -3 < AX < 0; \\ \ln  X + A^2  & \text{при } AX < -6 \text{ или } AX > 4, A \text{ и } X \text{ даны.} \end{cases}$

N	Формула для вычислений
10.	$Y = \begin{cases} \sqrt{A^3 + \text{Tg} AX} & \text{при } -6.8 \leq AX \leq -1.3 \text{ или } 0 \leq AX \leq 5.4; \\ A^2 - \text{Cos}^2 X^2 & \text{при } -1.3 < AX < 0; \\ \text{Ln} X^3 + A^2  & \text{при } AX < -6.8 \text{ или } AX > 5.4, A \text{ и } X \text{ даны.} \end{cases}$
11.	$Y = \begin{cases} \sqrt{\text{Sin} A^2 X} & \text{при } -10 \leq AX \leq -7 \text{ или } 0 \leq AX \leq 5.4; \\ A\sqrt{\text{Cos} X^3} & \text{при } -7 < AX < 0; \\ A + \text{Ln} AX  & \text{при } AX < -10 \text{ или } AX > 7, A \text{ и } X \text{ даны.} \end{cases}$
12.	$Y = \begin{cases} \sqrt{A^2 + \text{Ln} AX } & \text{при } -15 \leq AX \leq -7 \text{ или } 0 \leq AX \leq 10; \\ X + \sqrt{\text{Sin} A^3} & \text{при } -7 < AX < 0; \\ A + \text{Tg}^4 AX  & \text{при } AX < -15 \text{ или } AX > 10, A \text{ и } X \text{ даны.} \end{cases}$
13.	$Y = \begin{cases} \sqrt{\text{Ln} A^2 + X^3 } & \text{при } -2 \leq AX \leq 2 \text{ или } 4 \leq AX \leq 6; \\ X^3 e^{-AX} & \text{при } -7 < AX < 0; \\ \sqrt{A +  AX } & \text{при } AX < -2 \text{ или } AX > 6, A \text{ и } X \text{ даны.} \end{cases}$
14.	$Y = \begin{cases} \sqrt{X^2 + \text{Cos} AX} & \text{при } -11 \leq AX \leq -5 \text{ или } 0 \leq AX \leq 5; \\ 1 + A\sqrt{\text{Ctg} X^3} & \text{при } -5 < AX < 0; \\ A + \text{Ln} AX  & \text{при } AX < -11 \text{ или } AX > 5, A \text{ и } X \text{ даны.} \end{cases}$
15.	$Y = \begin{cases} \sqrt{1 + A\text{Tg} X^2} & \text{при } -15 \leq AX \leq 0 \text{ или } 3 \leq AX \leq 5; \\ \text{Ln} A + X  & \text{при } 0 < AX < 3; \\ A^2 + X^3 - 5 & \text{при } AX < -15 \text{ или } AX > 5, A \text{ и } X \text{ даны.} \end{cases}$

## Тема 5. Использование операторов циклов в методах

### Часть 1. Вычисление в методах конечных сумм и произведений

**Пример задания.** Ввести положительные числа  $x$  (с плавающей точкой) и  $n$  (целое). Выполнить проверку корректности введенных параметров (их неотрицательности). Трижды определить значение выражения:



$$S(x, n) = \sum_{i=1}^n \frac{i^2}{\sqrt{x+i}},$$

используя в трех различных методах по одному оператору цикла (for(), while(), do while()).

Замечание.

Хотя бы один из трех методов должен быть экземплярным, и один статическим.

**Пример программы.**

```

Main.java
1 import java.util.Scanner;
2
3 public class Main
4 {
5     //статический метод
6     static double forSum(double x, int n) {
7         double s = 0;
8         for (int i = 1; i <= n; i++) {
9             s = s + i * i / Math.sqrt(i + x);
10        }
11        return s;
12    }
13    //статический метод
14    static double whileSum(double x, int n) {
15        double s = 0;
16        int i = 1;
17        while(i <= n) {
18            s = s + i * i / Math.sqrt(i + x);
19            i++;
20        }
21        return s;
22    }
23    //экземплярный метод
24    double doSum(double x, int n) {
25        double s = 0;
26        int i = 1;
27        do {
28            s = s + i * i / Math.sqrt(i + x);
29            i++;
30        }

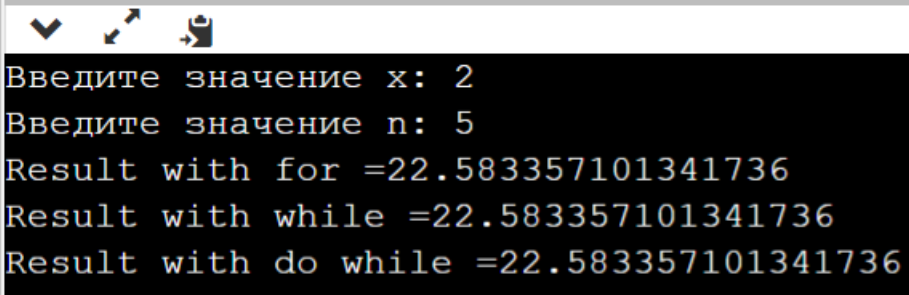
```

```

31     while(i <= n);
32     return s;
33 }
34
35 public static void main(String[] args) {
36     Scanner objIn = new Scanner(System.in);
37     double x;
38     int n;
39     while (true) {
40         System.out.print("Введите значение x: ");
41         x = objIn.nextDouble();
42         System.out.print("Введите значение n: ");
43         n = objIn.nextInt();
44         if (x > 0 && n > 0) break;
45         System.out.print("Некорректные данные!!!" +
46             "\nПопробуйте еще раз!!!");
47     }
48     objIn.close();
49     Main obj = new Main();
50     System.out.print("Result with for =" +
51         forSum(x, n) +
52         "\nResult with while =" +
53         whileSum(x, n) +
54         "\nResult with do while =" +
55         obj.doSum(x, n));
56 }
57 }

```

### Результат работы программы:



```

Введите значение x: 2
Введите значение n: 5
Result with for =22.583357101341736
Result with while =22.583357101341736
Result with do while =22.583357101341736

```

## Варианты индивидуальных заданий.

Используя три различных цикла, определить значение суммы или произведения:

N	Формула	N	Формула
1.	$S(n) = \sum_{i=2}^n \ln(i);$	9.	$S(x, n) = \sum_{i=1}^n (x \cdot i - 1)^3;$
2.	$P(x, n) = \prod_{i=1}^n (i^2 + x);$	10.	$P(x, n) = \prod_{i=1}^n (2 \cdot x + \sqrt{i});$
3.	$S(n) = \sum_{i=1}^n (i + 1)^2;$	11.	$P(x, n) = \prod_{i=1}^n (i + \sqrt{2 \cdot x});$
4.	$P(x, n) = \prod_{i=1}^n (i + x);$	12.	$S(x, n) = \sum_{i=1}^n \frac{i+x}{(i+1)^2};$
5.	$S(x, n) = \sum_{i=1}^n (i^2 + x);$	13.	$P(x, n) = \prod_{i=1}^n \left( x + \sqrt{\frac{1}{i}} \right);$
6.	$S(x, n) = \sum_{i=1}^n (i^3 + i \cdot x);$	14.	$S(x, n) = \sum_{i=1}^n \frac{2}{(i+x)^3};$
7.	$P(x, n) = \prod_{i=1}^n \left( \frac{i}{2} + x \right);$	15.	$P(x, n) = \prod_{i=1}^n \ln(i + 2 \cdot x).$
8.	$P(x, n) = \prod_{i=1}^n \left( \frac{1}{\sqrt{2 \cdot i}} + x \right);$	16.	$S(n) = \sum_{i=1}^n \frac{3^{i+1}}{i^3}$

## Часть 2. Цепочка вызовов методов. Табулирование значений математической функции

**Пример задания.** С помощью метода задать математическую функцию  $f(x)$  в соответствии с заданием (например,  $f(x) = (x + 1)^2$ ). Протабулировать ее значения на интервале  $[0, 3]$  с шагом 0.2, используя в трех различных методах три оператора цикла `for()` (на интервале  $[0, 1[$ ), `while()` (на интервале  $[1, 2[$ ), `do while()` (на интервале  $[2, 3]$ ), указав при этом значения функции с нечетной и четной целой частью.

Для вывода строк таблицы с требуемой информацией использовать методы `printFor()`, `printWhile()` и `printDo()`, выполняющих табулирование с помощью соответствующего оператора цикла.

### Замечание.

Хотя бы один из этих методов должен быть экземплярным, и хотя бы один статическим.



## Пример программы.

Main.java

```
1 public class Main
2 {
3     //статический метод
4     static double UserFunction (double x) {
5         return (x + 1) * (x + 1);
6     }
7     //статический метод
8     static String informationString(double x) {
9         //метод format() создает из форматированной строки
10        //после подстановки значений параметров строку
11        //(тип String)
12        String result = String.format("\nx = %4.2f" +
13                                     "\tf = %7.3f" +
14                                     "\tInteger part of" +
15                                     " f(x) %d is ",
16                                     x, UserFunction(x),
17                                     (int) UserFunction(x));
18        if (((int) UserFunction(x)) % 2 != 0) {
19            result = result + "odd.";
20        }
21        else result = result + "even.";
22        return result;
23    }
24    //статический метод
25    static void printFor(double a, double b, double step) {
26        for(double x = a; x < b; x+= step) {
27            System.out.print(informationString(x));
28        }
29    }
30    //статический метод
31    static void printWhile(double a, double b, double step) {
32        double x = a;
33        while(x < b) {
34            System.out.print(informationString(x));
35            x+= step;
36        }
37    }
38    //экземплярный метод
39    void printDo(double a, double b, double step) {
40        double x = a;
41        do {
42            System.out.print(informationString(x));
43            x+= step;
44        }
45        while(x < b + step);
```

```

46     }
47
48     public static void main(String[] args) {
49         printFor(0, 1, 0.2);
50         printWhile(1, 2, 0.2);
51         Main obj = new Main();
52         obj.printDo(2, 3, 0.2);
53     }
54 }

```

### Результат работы программы:

			input
x = 0.00	f = 1.000	Integer part of f(x)	1 is odd.
x = 0.20	f = 1.440	Integer part of f(x)	1 is odd.
x = 0.40	f = 1.960	Integer part of f(x)	1 is odd.
x = 0.60	f = 2.560	Integer part of f(x)	2 is even.
x = 0.80	f = 3.240	Integer part of f(x)	3 is odd.
x = 1.00	f = 4.000	Integer part of f(x)	4 is even.
x = 1.20	f = 4.840	Integer part of f(x)	4 is even.
x = 1.40	f = 5.760	Integer part of f(x)	5 is odd.
x = 1.60	f = 6.760	Integer part of f(x)	6 is even.
x = 1.80	f = 7.840	Integer part of f(x)	7 is odd.
x = 2.00	f = 9.000	Integer part of f(x)	9 is odd.
x = 2.00	f = 9.000	Integer part of f(x)	9 is odd.
x = 2.20	f = 10.240	Integer part of f(x)	10 is even.
x = 2.40	f = 11.560	Integer part of f(x)	11 is odd.
x = 2.60	f = 12.960	Integer part of f(x)	12 is even.
x = 2.80	f = 14.440	Integer part of f(x)	14 is even.
x = 3.00	f = 16.000	Integer part of f(x)	16 is even.

### Варианты индивидуальных заданий.

Используя три различных цикла, определить значение суммы или произведения:

N	$f(x)$
1.	$\frac{2 \cdot x + 1}{x^3 + 2 \cdot x^2 + x + 2}$
2.	$\sqrt{3^{-2 \cdot x^2} + 2^{4 \cdot x}}$
3.	$4 \cdot \frac{\sin(x^3 + 2 \cdot x^2 + x + 2)}{x^2 + 2}$
4.	$\begin{cases} (x + 4)^2, & \text{при }  \sin(x)  > 0.5 \\ x, & \text{при }  \sin(x)  \leq 0.5 \end{cases}$

N	$f(x)$
5.	$\sqrt{2^{3 \cdot x} + 5 \cdot x^2 + x - 1};$
6.	$ \sin(x^2 + 3) $
7.	$e^{3 \cdot x^2 +  \sin(x) };$
8.	$\sin\left(\frac{x^4 + x + 7}{x + 5}\right)$

N	$f(x)$	N	$f(x)$
9.	$\begin{cases} \sqrt{ x } + 4, & \text{при }  \cos(x)  > 0.5 \\ x - 1, & \text{при }  \cos(x)  \leq 0.5 \end{cases}$	13.	$\sqrt{e^{x^2} + e^{-x^2}}$
10.	$\sin(4 \cdot \operatorname{tg}(x))$	14.	$\sqrt[4]{\left  \frac{2}{(1+x)^3} \right }$
11.	$\sin^2(\operatorname{ctg}( x  + 2))$	15.	$\frac{3^{\sin(x)}}{ x^3  + 1}$
12.	$\operatorname{tg}( x^5  + 2)$	16.	$\ln( \sin(x)  + 2 \cdot x)$

## Тема 6. Создание класса-библиотеки методов для циклических вычислений с известным числом повторений

**Пример задания.** Ввести положительные числа  $x$  (с плавающей точкой) и  $n$  (целое). Выполнить проверку корректности введенных параметров (их неотрицательности). Трижды определить значение выражения:  $(x + 1)^n$  используя его в трех различных методах по одному оператору цикла (`for()`, `while()`, `do while()`).

### Пример программы.

```

Main.java
1 import java.util.Scanner;
2 //класс-библиотека
3 class methodLibrary {
4     //статический метод
5     static double forProduct(double x, int n) {
6         double product = 1;
7         for (int i = 1; i <= n; i++) {
8             product = product * (x + 1);
9         }
10        return product;
11    }
12    //статический метод
13    static double whileProduct(double x, int n) {
14        double product = 1;
15        int i = 1;

```

```

16 ▾      while( i <= n ) {
17          product = product * (x + 1);
18          i++;
19      }
20      return product;
21  }
22  //экземплярный метод
23 ▾  double doProduct(double x, int n) {
24      double product = 1;
25      int i = 1;
26 ▾  do {
27          product = product * (x + 1);
28          i++;
29      }
30      while( i <= n );
31      return product;
32  }
33 }
34
35 public class Main
36 {
37 ▾  public static void main(String[] args) {
38      Scanner objIn = new Scanner(System.in);
39      double x;
40      int n;
41 ▾  while (true) {
42      System.out.print("Введите значение x: ");
43      x = objIn.nextDouble();
44      System.out.print("Введите значение n: ");
45      n = objIn.nextInt();
46      if (x > 0 && n > 0) break;
47 ▾  System.out.print("Некорректные данные!!!" +
48                  "\nПопробуйте еще раз!!!");
49      }
50      objIn.close();
51      methodLibrary obj = new methodLibrary();
52 ▾  System.out.print("Result with for =" +
53                  methodLibrary.forProduct(x, n) +
54                  "\nResult with while =" +
55                  methodLibrary.whileProduct(x, n)+
56                  "\nResult with do while =" +
57                  obj.doProduct(x, n));
58      }
59 }

```

## Результат работы программы:

```
✓ ↗ 🗑️
Введите значение x: 2
Введите значение n: 5
Result with for =243.0
Result with while =243.0
Result with do while =243.0
```

## Часть 1. Создание экземплярного метода класса-библиотеки

**Варианты индивидуальных заданий.** Создать метод:

1. вычисляющий (возвращающий значение) факториала целого  $n$  ( $n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$ ), где  $n$  передается в метод как параметр;
2. вычисляющий (возвращающий)  $x^n$ , где  $x$  – число с плавающей точкой,  $n$  – целое число. Метод получает  $x$  и  $n$  как параметр;
3. вычисляющий  $(-1)^n \cdot n!$ , где целое  $n$  передается в метод как параметр;
4. вычисляющий  $2^{2 \cdot n + 1}$ , где целое  $n$  передается в метод как параметр;
5. вычисляющий (возвращающий значение) двойного факториала целого  $n$  (если  $n$  нечетное, то  $n!! = 1 \cdot 3 \cdot 5 \cdot \dots \cdot n$ , если  $n$  четное, то  $n!! = 1 \cdot 2 \cdot 4 \cdot \dots \cdot n$ ), где  $n$  передается в метод как параметр;
6. вычисляющий (возвращающий)  $(\sin(x))^{2 \cdot n}$ , где  $x$  – число с плавающей точкой,  $n$  – целое число. Метод получает  $x$  и  $n$  как параметр;
7. вычисляющий  $(-1)^n x^n$ , где целое  $n$  и вещественное  $x$  передается в метод как параметр;
8. вычисляющий  $3^{2 \cdot n}$ , где целое  $n$  передается в метод как параметр;
9. вычисляющий (возвращающий значение) произведения  $\left(1 + \frac{1}{1^2}\right) \cdot \left(1 + \frac{1}{2^2}\right) \cdot \dots \cdot \left(1 + \frac{1}{n^2}\right)$  для заданного целого  $n$ , передаваемого в метод как параметр;
10. вычисляющий (возвращающий)  $x^{3 \cdot n}$ , где  $x$  – число с плавающей точкой,  $n$  – целое число. Метод получает  $x$  и  $n$  как параметры;
11. вычисляющий  $(-1)^n x^{2 \cdot n}$ , где целое  $n$  и вещественное  $x$  передается в метод как параметр;
12. вычисляющий  $7^{3 \cdot n}$ , где целое  $n$  передается в метод как параметр;
13. вычисляющий (возвращающий значение) факториал целого  $(2 \cdot n + 1)$  ( $(2 \cdot n + 1)! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot (2 \cdot n + 1)$ ), где  $n$  передается в метод как параметр;

14. вычисляющий (возвращающий)  $(x + 5)^{2n+1}$ , где  $x$  – число с плавающей точкой,  $n$  – целое число. Метод получает  $x$  и  $n$  как параметр;
15. вычисляющий  $(-1)^n x^{2 \cdot n + 1}$ , где целое  $n$  и вещественное  $x$  передается в метод как параметр;
16. вычисляющий  $(2 \cdot n)^n$ .

## Часть 2. Создание статического метода класса-библиотеки

**Варианты индивидуальных заданий.** Дано натуральное  $n$ . Вычислить значение суммы  $S$ :

1.  $S = 1 + 2 + \dots + n$ ;
2.  $S = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$ ;
3.  $S = 1^2 + 2^2 + \dots + n^2$ ;
4.  $S = \frac{1}{1^3} + \frac{1}{2^3} + \frac{1}{3^3} + \dots + \frac{1}{n^3}$
5.  $S = 1 + 3 + 5 + \dots + n$ , где  $n$  – нечетное число;
6.  $S = \frac{1}{1^3} + \frac{1}{3^3} + \frac{1}{5^3} + \dots + \frac{1}{n^3}$ , где  $n$  – нечетное число;
7.  $S = 2 + 4 + 6 + \dots + n$ , где  $n$  – четное число;
8.  $S = \frac{1}{2^5} + \frac{1}{4^5} + \frac{1}{6^5} + \dots + \frac{1}{n^5}$ , где  $n$  – четное число;
9.  $S = 1 \cdot 2 + 2 \cdot 3 + 3 \cdot 4 + \dots + n \cdot (n + 1)$ ;
10.  $S = \frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \frac{1}{3 \cdot 4} + \dots + \frac{1}{n \cdot (n+1)}$ ;
11.  $S = 1 \cdot 3 + 3 \cdot 5 + 5 \cdot 7 + \dots + n \cdot (n + 2)$ , где  $n$  – нечетное число;
12.  $S = \frac{1}{1 \cdot 3} + \frac{1}{3 \cdot 5} + \frac{1}{5 \cdot 7} + \dots + \frac{1}{n \cdot (n+2)}$ , где  $n$  – нечетное число;
13.  $S = 2 \cdot 4 + 4 \cdot 6 + 6 \cdot 8 + \dots + n \cdot (n + 2)$ , где  $n$  – четное число;
14.  $S = \frac{1}{2 \cdot 4} + \frac{1}{4 \cdot 6} + \frac{1}{6 \cdot 8} + \dots + \frac{1}{n \cdot (n+2)}$ , где  $n$  – четное число;
15.  $S = \sin(1) + \sin(2) + \dots + \sin(2 \cdot n + 1)$ ;
16.  $S = \frac{1}{\sin(1)} + \frac{1}{\sin(2)} + \frac{1}{\sin(3)} + \dots + \frac{1}{\sin(n)}$ ;

## Тема 7. Разработка класса, содержащего методы реализующие итерационные вычисления

### Часть 1. Вычисление с помощью статического метода бесконечной суммы ряда с заданной точностью

**Обобщенная формулировка.** Даны действительные числа  $x, \varepsilon$  ( $\varepsilon > 0$ ). Вычислить в *статическом* методе «внешнего» класса по отношению к классу Main с заданной точностью  $\varepsilon$  приближенное значение бесконечной суммы (математической функции) и сравнить его со значением, соответствующего стандартного статического метода класса Math.

**Пример задания.** Рассмотрим пример итерационного вычисления с помощью разложения в бесконечный ряд. Функции  $e^x$  на интервале  $x \in (-1,1)$  с точностью  $\varepsilon$ :

$$1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!} + \dots$$

Критерием остановки суммирования является условие малости очередного прибавляемого члена ряда по отношению к введенному значению точности:

$$\left| \frac{x^n}{n!} \right| \leq \varepsilon$$

### Пример программы.

```
Main.java
1 import java.util.Scanner;
2 //«внешний» класс-библиотека методов
3 class IterativeCalculator {
4     //статический метод
5     static double expUser(double x, double epsilon) {
6         double sum = 1, term = 1;
7         for (int i = 1; Math.abs(term) > epsilon; i++) {
8             term = term * x / i;
9             sum = sum + term;
10        }
11        return sum;
12    }
13 }
```

```

14
15 public class Main
16 {
17     public static void main(String[] args) {
18         Scanner objIn = new Scanner(System.in);
19         double x, epsilon;
20         while (true) {
21             System.out.print("Введите x: ");
22             x = objIn.nextDouble();
23             System.out.print("Введите точность: ");
24             epsilon = objIn.nextDouble();
25             if(Math.abs(x) < 1 && Math.abs(epsilon) < 0.05)
26                 break;
27             System.out.print("Некорректные данные!!!" +
28                 "\nПопробуйте еще раз!!!");
29         }
30         objIn.close();
31
32         System.out.print("Значение вычисленной суммы = "+
33             IterativeCalculator.expUser(x, epsilon) +
34             "\nЗначение встроенного метода = " +
35             Math.exp(x));
36     }
37 }

```

### Результаты работы программы:

```

Введите x: 0.5
Введите точность: 0.005
Значение вычисленной суммы = 1.6484375
Значение встроенного метода = 1.6487212707001282

```

### Варианты индивидуальных заданий.

№	$\sum$	Точное значение
1	$1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + (-1)^m \frac{x^{2m}}{(2m)!} + \dots$	$\cos x, x \in R$
2	$x + \frac{x^3}{3!} + \frac{x^5}{5!} + \dots + \frac{x^{2m-1}}{(2m-1)!} + \dots$	$\operatorname{sh} x, x \in R$
3	$1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots + \frac{x^{2m}}{(2m)!} + \dots$	$\operatorname{ch} x, x \in R$
4	$x - \frac{x^2}{2} + \frac{x^3}{3} + \dots + (-1)^{n-1} \frac{x^n}{n} + \dots$	$\ln(1+x),$ $x \in ]-1;1]$



№	$\Sigma$	Точное значение
5	$x - \frac{x^3}{3} + \frac{x^5}{5} + \dots + (-1)^{m-1} \frac{x^{2m-1}}{2m-1} + \dots$	$\operatorname{arctg}x$ , $x \in [-1;1]$
6	$-x - \frac{x^2}{2} - \frac{x^3}{3} - \dots - \frac{x^n}{n} - \dots$	$\ln(1-x)$ , $x \in [-1;1[$
7	$1 - x + x^2 - \dots + (-1)^n x^n + \dots$	$\frac{1}{1+x}$ , $x \in ]-1;1[$
8	$1 + x + x^2 - \dots + x^n + \dots$	$\frac{1}{1-x}$ , $x \in ]-1;1[$
9	$1 + \alpha x + \frac{\alpha(\alpha-1)}{2!} x^2 + \dots + \frac{\alpha(\alpha-1)\dots(\alpha-n+1)}{n!} x^n + \dots$	$(1+x)^\alpha$ , $x \in ]-1;1[$
10	$2 \left( x + \frac{x^3}{3} + \frac{x^5}{5} + \dots + \frac{x^{2n-1}}{2n-1} + \dots \right)$	$\ln \frac{1+x}{1-x}$ , $x \in ]-1;1[$
11	$1 - \frac{1}{2}x + \frac{3}{8}x^2 - \frac{5}{16}x^3 + \dots + (-1)^{n-1} \frac{(2n-3)!!}{(2n)!!} x^n + \dots$	$\frac{1}{\sqrt{1+x}}$ , $x \in [-1;1]$

## Часть 2. Определение предела сходящейся последовательности в экземплярном методе

**Обобщенная формулировка.** Вычислить предел сходящейся последовательности  $\{a_n\}_{n=1}^{\infty}$ . Вычисления прекратить, если выполнено условие  $|a_{n+1} - a_n| < \varepsilon$ . После окончания вычислений в качестве значения предела вывести на экран значение  $a_n$ .

Рассмотрим пример вычисления предела последовательности

$$a_n = \frac{n}{\sqrt{n^2+1} + \sqrt{n^2-1}} \quad (\text{т.е. вычислить приближенное значение } \lim_{n \rightarrow \infty} \frac{n}{\sqrt{n^2+1} + \sqrt{n^2-1}}).$$

### Пример программы.

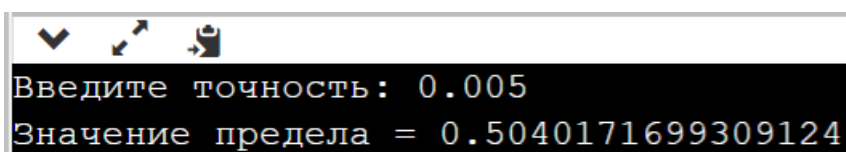
```
Main.java
1 import java.util.Scanner;
2 //«внешний» класс-библиотека методов, определяющих
3 //«что итерировать»
```

```

4 class UserLibrary {
5     //экземплярный метод
6     double function(int n) {
7         return n / (Math.sqrt(n * n + 1) + Math.sqrt(n * n - 1));
8     }
9 }
10 //«внешний» класс-библиотека методов, определяющих «как
11 //итерировать»
12 class IterativeCalculator {
13     //экземплярный метод
14     double sequence(double epsilon) {
15         UserLibrary object = new UserLibrary();
16         double current, next;
17         int i = 1;
18         do {
19             current = object.function(i);
20             next = object.function(i + 1);
21             i++;
22         }
23         while (Math.abs(current - next) > epsilon);
24         return current;
25     }
26 }
27
28 public class Main
29 {
30     public static void main(String[] args) {
31         Scanner objIn = new Scanner(System.in);
32         double epsilon;
33         while (true) {
34             System.out.print("Введите точность: ");
35             epsilon = objIn.nextDouble();
36             if(Math.abs(epsilon) < 0.05) break;
37             System.out.print("Некорректные данные!!!" +
38                 "\nПопробуйте еще раз!!!");
39         }
40         objIn.close();
41         IterativeCalculator obj = new IterativeCalculator();
42         System.out.print("Значение предела = " +
43             obj.sequence(epsilon));
44     }
45 }

```

**Результат работы программы:**



```

Введите точность: 0.005
Значение предела = 0.5040171699309124

```

**Варианты индивидуальных заданий.** Вычислить предел последовательности  $\lim_{n \rightarrow \infty} a_n$ :

1.  $a_n = \frac{3 \cdot n^2 - 7 \cdot n + 1}{2 - 5 \cdot n - 6 \cdot n^2} (n = 0, 2, \dots);$

2.  $a_n = \frac{2}{n} (n = 1, 2, \dots);$

3.  $a_n = \frac{n-1}{n^2} (n = 1, 2, \dots);$

4.  $a_n = \frac{(-1)^n}{n+1} (n = 0, 2, \dots);$

5.  $a_n = \frac{1}{3^n} (n = 0, 2, \dots);$

6.  $a_n = \frac{1}{\sqrt{n+5}} (n = 0, 2, \dots);$

7.  $a_n = \frac{1}{n\sqrt{2}} (n = 1, 2, \dots);$

8.  $a_n = \frac{1}{\sqrt[3]{n^2+4n-2}} (n = 1, 2, \dots);$

9.  $a_n = \sin\left(\frac{1}{n+3}\right) (n = 0, 2, \dots).$

## Тема 8. Создание класса, содержащего методы, реализующие рекурсию

### Часть 1. Статические методы, реализующие рекурсию

Реализовать рекурсивный алгоритм с помощью статического метода «внешнего» класса.

**Пример задания.** Написать статический метод «внешнего» класса RecursiveCalculator, рекурсивно вычисляющий факториал натурального числа  $n = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$  ( $n!$ ) с использованием тернарной операции.

#### Пример программы.

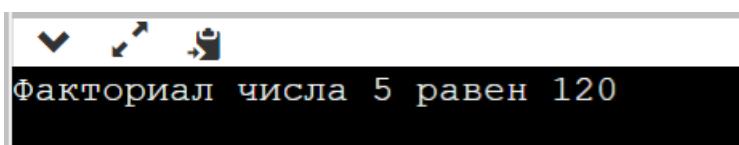
```
Main.java
1 import java.util.Scanner;
2
3 class RecursiveCalculator {
```

```

4 //статический метод
5 public static long factorial(int n) {
6     return n == 1 ? 1 : n * factorial(n - 1);
7 }
8 }
9
10 public class Main
11 {
12     public static void main(String[] args) {
13         System.out.print("Факториал числа 5 равен "+
14             RecursiveCalculator.factorial(5));
15     }
16 }

```

**Результат работы программы:**



**Варианты индивидуальных заданий.** Рекурсивно определить значение суммы или произведения:

N	Формула
9.	$S(n) = \sum_{i=2}^n \ln(i);$
10.	$P(x, n) = \prod_{i=1}^n (i^2 + x);$
11.	$S(n) = \sum_{i=1}^n (i + 1)^2;$
12.	$P(x, n) = \prod_{i=1}^n (i + x);$
13.	$S(x, n) = \sum_{i=1}^n (i^2 + x);$
14.	$S(x, n) = \sum_{i=1}^n (i^3 + i \cdot x);$
15.	$P(x, n) = \prod_{i=1}^n \left(\frac{i}{2} + x\right);$
16.	$P(x, n) = \prod_{i=1}^n \left(\frac{1}{\sqrt{2} \cdot i} + x\right);$

N	Формула
17.	$S(x, n) = \sum_{i=1}^n (x \cdot i - 1)^3;$
18.	$P(x, n) = \prod_{i=1}^n (2 \cdot x + \sqrt{i});$
19.	$P(x, n) = \prod_{i=1}^n (i + \sqrt{2 \cdot x});$
20.	$S(x, n) = \sum_{i=1}^n \frac{i+x}{(i+1)^2};$
21.	$P(x, n) = \prod_{i=1}^n \left(x + \sqrt{\frac{1}{i}}\right);$
22.	$S(x, n) = \sum_{i=1}^n \frac{2}{(i+x)^3};$
23.	$P(x, n) = \prod_{i=1}^n \ln(i + 2 \cdot x).$
24.	$S(n) = \sum_{i=1}^n \frac{3^{i+1}}{i^3}$

## Часть 2. Экземплярные методы, реализующие рекурсию

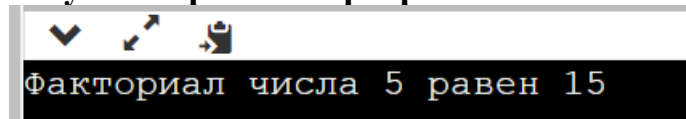
**Обобщенная формулировка задания.** Реализовать рекурсивный алгоритм с помощью экземплярного метода «внешнего» класса.

**Пример задания.** Написать экземплярный метод «внешнего» класса RecursiveCalculator, выполняющий рекурсивное суммирование целых чисел до заданного  $n$ :  $1 + 2 + \dots + n$ .

### Пример программы.

```
Main.java
1 import java.util.Scanner;
2
3 class RecursiveCalculator {
4     //экземплярный метод
5     public int sum(int n) {
6         return n == 0 ? 0 : n + sum(n - 1);
7     }
8 }
9
10 public class Main
11 {
12     public static void main(String[] args) {
13         RecursiveCalculator obj = new RecursiveCalculator();
14         System.out.print("Факториал числа 5 равен "+
15             obj.sum(5));
16     }
17 }
```

### Результат работы программы:



```
Факториал числа 5 равен 15
```

**Варианты индивидуальных заданий.** Во «внешнем» классе RecursiveCalculator написать экземплярный рекурсивный метод, вычисляющий (возвращающий в качестве значения):

1.  $(-1)^n \cdot n!$ , где целое  $n$  передается в метод как параметр;
2.  $2^{2 \cdot n + 1}$ , где целое  $n$  передается в метод как параметр;
3. двойной факториал целого  $n$  (если  $n$  нечетное, то  $n!! = 1 \cdot 3 \cdot 5 \cdot \dots \cdot n$ , если  $n$  четное, то  $n!! = 1 \cdot 2 \cdot 4 \cdot \dots \cdot n$ ), где  $n$  передается в метод как параметр;

4.  $(\sin(x))^{2 \cdot n}$ , где  $x$  – число с плавающей точкой,  $n$  – целое число. метод получает  $x$  и  $n$  как параметр;
5.  $(-1)^n x^n$ , где целое  $n$  и вещественное  $x$  передается в метод как параметр;
6.  $3^{2 \cdot n}$ , где целое  $n$  передается в метод как параметр;
7. произведение  $\left(1 + \frac{1}{1^2}\right) \cdot \left(1 + \frac{1}{2^2}\right) \cdot \dots \cdot \left(1 + \frac{1}{n^2}\right)$  для заданного целого  $n$ , передаваемого в метод как параметр;
8.  $x^{3 \cdot n}$ , где  $x$  – число с плавающей точкой,  $n$  – целое число. Метод получает  $x$  и  $n$  как параметр;
9.  $(-1)^n x^{2 \cdot n}$ , где целое  $n$  и вещественное  $x$  передается в метод как параметр;
10.  $7^{3 \cdot n}$ , где целое  $n$  передается в метод как параметр;
11. вычисляющий (возвращающий значение) факториал целого  $(2 \cdot n + 1)$  ( $(2 \cdot n + 1)! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot (2 \cdot n + 1)$ ), где  $n$  передается в метод как параметр;
12.  $(x + 5)^{2n+1}$ , где  $x$  – число с плавающей точкой,  $n$  – целое число. Метод получает  $x$  и  $n$  как параметр;
13.  $(-1)^n x^{2 \cdot n+1}$ , где целое  $n$  и вещественное  $x$  передается в метод как параметр;
14.  $(2 \cdot n)^n$ ;
15.  $x^n$ , где  $x$  – число с плавающей точкой,  $n$  – целое число. Метод получает  $x$  и  $n$  как параметр;

## Тема 9. Создание класса, содержащего методы обработки строк

**Обобщенная формулировка задания.** Библиотека (класс `Library`) должна содержать *четыре* метода:

- первый статический из части 1 списка индивидуальных заданий;
- второй статический из части 2 списка индивидуальных заданий
- третий экземплярный из части 3 списка индивидуальных заданий;
- четвертый экземплярный из части 4 списка индивидуальных заданий.

**Пример задания с двумя методами.** Класс `Library` содержит:

1. Статический метод, удаляющий нецифровые символы из строки, представляющей целое число и переводящего его в тип `int`. При написании данного метода необходимо помнить, что сам перевод

строки состоящей только из цифр в целое число осуществляется соответствующими методами (`parseInt()`, `parseByte()` и др.) соответствующих классов-обертки. Поскольку речь идет об удалении некорректных символов из записи, то проще всего это сделать, преобразовав строку в объект класса `StringBuffer`. Он имеет метод `delete()`, который поможет кратчайшим образом решить поставленную задачу.

2. Экземплярный метод, осуществляющий подготовку строки к переводу в формат числа с плавающей точкой. Особенностью метода является, то, что на первом этапе все запятые в записи числа заменяются на точки, а далее все запятые, находящиеся справа от самой первой (слева по порядку записи) точки, удаляются. Следующая часть алгоритма полностью совпадает с алгоритмом для перевода записи целого числа, но с тем отличием, что в строку с допустимыми символами добавлена «точка». В завершении алгоритма для исправленной строки должен использоваться метод `parseDouble()` класса-обертки `Double`.

### Пример программы.

```
Main.java
1 import java.util.Scanner;
2 import java.util.Locale;
3
4 class Library {
5     //статический метод подготовка к переводу строки в
6     //целое число
7     static String intDelInvalid(String str) {
8         StringBuffer a = new StringBuffer(str);
9         String validChar = "0123456789";
10        //инициализация индекса символа введенной
11        //строки в зависимости от знака числа
12        int i = (a.charAt(0) == '-' ? 1 : 0);
13        for(; i < a.length(); ){
14            if(validChar.indexOf(a.charAt(i)) < 0) {
15                a.delete(i, i + 1);
16            }
17            else i++;
18        }
19        return a.toString();
20    }
21    //экземплярный метод подготовки к переводу в число
22    //Double
23    String doubleDelInvalid(String str) {
24        //предварительная замена всех запятых на точку
25        StringBuffer a =
```

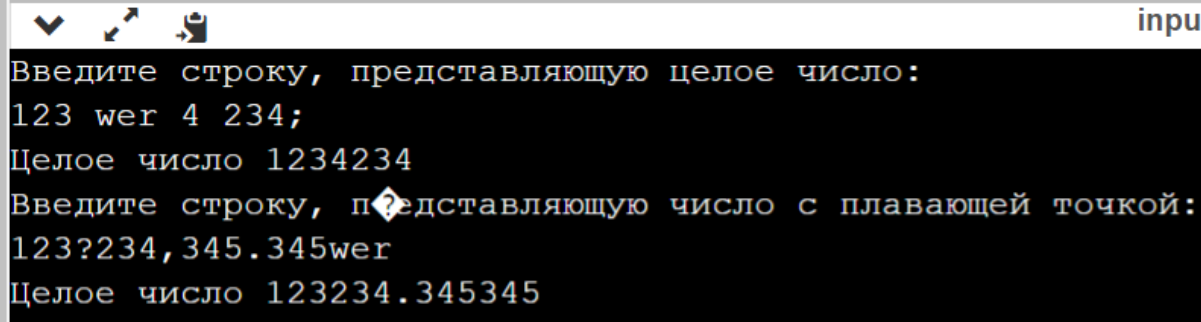
```

26         new StringBuffer(str.replace(',', '.'));
27 //поиск первой точки слева и удаление всех точек
28 //справа от нее
29 for(int i = a.toString().lastIndexOf('.');
30     a.toString().indexOf('.') < i; ){
31     a.delete(i, i+1);
32     i = a.toString().lastIndexOf('.');
33 }
34 //строка допустимых символов дополняется точкой
35 String validChar = ".0123456789";
36 //повторение алгоритма удаления недопустимых
37 //символов из предыдущего метода
38 int i = (a.charAt(0) == '-' ? 1 : 0);
39 for(; i < a.length(); ){
40     if(validChar.indexOf(a.charAt(i)) < 0) {
41         a.delete(i, i + 1);
42     }
43     else i++;
44 }
45 return a.toString();
46 }
47 }
48
49 public class Main
50 {
51     public static void main (String args[]) {
52         Scanner objIn = new Scanner(System.in);
53         Locale locale =
54             new Locale.Builder().setLanguage("en")
55                 .setRegion("US").build();
56         objIn.useLocale(locale);
57         System.out.println("Введите строку, представляющую "+
58             "целое число: ");
59         // ввод строки
60         String str = objIn.nextLine();
61
62         System.out.println("Целое число "+
63             Integer.parseInt(Library.intDelInvalid(str)));
64         System.out.println("Введите строку, представляющую "+
65             "число с плавающей точкой: ");
66         // ввод строки
67         str = objIn.nextLine();
68         objIn.close();
69         Library obj = new Library();
70         System.out.println("Целое число "+
71             Double.parseDouble(obj.doubleDelInvalid(str)));
72     }
73 }

```



## Результат работы программы:



```
inpu
Введите строку, представляющую целое число:
123 wer 4 234;
Целое число 1234234
Введите строку, представляющую число с плавающей точкой:
123?234,345.345wer
Целое число 123234.345345
```

### Замечания:

- Особенностью работы со строками Java является постоянная конвертация строки то в объект класса *StringBuffer*, то возвращение в класс *String*. Это делается потому, что разные классы имеют разные методы, которыми следует пользоваться в разных частях алгоритма.

## Часть 1. Задачи определения числовых характеристик заданной строки

**Варианты индивидуальных заданий.** Дана строка латинских символов и разделителей (пробелы и знаки препинания). С помощью методов:

1. Посчитать количество разделителей (пробелы и знаки препинания). Создать вспомогательную строку из разделителей.
2. Определить номер по порядку первого слова, начинающегося с гласной буквы. Создать вспомогательные строки из гласных букв и разделителей.
3. Посчитать количество заглавных согласных. Создать вспомогательную строку из заглавных согласных букв.
4. Определить номер последнего слова, начинающегося с согласной. Создать вспомогательную строку из согласных букв и разделителей.
5. Посчитать количество заглавных гласных в строке. Создать вспомогательную строку из заглавных гласных букв.
6. Определить длину первого слова, начинающегося с согласной. Создать вспомогательную строку из согласных букв и разделителей.
7. Посчитать количество слов в строке. Создать вспомогательную строку из разделителей.
8. Посчитать наибольшее количество подряд идущих гласных в строке. Создать вспомогательную строку из гласных.

9. В строке посчитать наибольшее количество групп знаков из подряд идущих разделителей. Создать вспомогательную строку разделителей.
10. В строке посчитать количество строчных согласных. Создать вспомогательную строку строчных согласных.
11. В строке посчитать сколько вхождений в строку имеет комбинация знаков «согласная буква-разделитель-согласная буква». Создать вспомогательные строки согласных букв и разделителей.
12. Посчитать количество строчных гласных. Создать вспомогательную строку строчных гласных.
13. Посчитать сколько вхождений в строку имеет комбинация букв «гласная буква-разделитель-гласная буква». Создать вспомогательные строки гласных и разделителей.
14. Определить среднюю длину слов в строке. Создать вспомогательную строку разделителей.
15. Определить количество слов длина, которых больше/меньше заданного числа. Создать вспомогательную строку разделителей.
16. Определить количество слов начинающихся и заканчивающихся одной и той же буквой. Создать вспомогательную строку разделителей.
17. Определить длину самого короткого/длинного слова в строке. Создать вспомогательную строку разделителей.
18. Определить номер по порядку самого длинного/короткого слова в строке. Создать вспомогательную строку разделителей.
19. Посчитать наибольшее количество подряд идущих согласных в строке. Создать вспомогательную строку из согласных.

## **Часть 2. Удаление (забой) символа в строке сдвигом влево**

**Варианты индивидуальных заданий.** Дана строка (набор символов и разделителей). Удалить в методе определенное множество символов в строке сдвигом влево.

1. Удалить из строки все пробелы, сдвигая строку при удалении каждого заданного знака влево на один символ (забой или `BackSpace`).
2. Удалить разделители сдвигом влево между словами, первое из которых заканчивается на гласную, а следующее начинается на согласную.

3. Удалить из строки все запятые, сдвигая строку при удалении каждого заданного знака влево на один символ (забой или BackSpace).
4. Удалить из строки все точки, сдвигая строку при удалении каждого заданного знака влево на один символ (забой или BackSpace).
5. Удалить из строки все восклицательные знаки, сдвигая строку при удалении каждого заданного знака влево на один символ (забой или BackSpace).
6. Удалить из строки сдвигом влево все повторяющиеся гласные (первый символ в последовательности повторяющихся гласных оставить).
7. Удалить из строки сдвигом влево все повторяющиеся согласные (первый символ в последовательности повторяющихся согласных оставлять).
8. Удалить из строки все двоеточия, сдвигая строку при удалении каждого заданного знака влево на один символ (забой или BackSpace).
9. Удалить из строки все точки с запятой, сдвигая строку при удалении каждого заданного знака влево на один символ (забой или BackSpace).
10. Удалить сдвигом влево все символы из строки между вторым и последним пробелами. Пробел между словами может остаться только один.
11. Удалить из строки сдвигом влево все символы между первой и последней запятыми (остается только одна запятая). Символ конца '\0' строки также должен быть сдвинут влево при удалении заданного фрагмента.
12. Удалить все знаки из строки сдвигом влево между второй и предпоследней гласной буквой (остается только начальная гласная, определяющая строку для удаления).
13. Удалить сдвигом влево первое слово в строке.
14. Удалить из строки сдвигом влево все самые длинные слова.
15. Удалить из строки сдвигом влево все самые короткие слова.

### Часть 3. Циклическая перестановка символов в подстроке

**Варианты индивидуальных заданий.** Дана строка (набор символов и разделителей). Циклически переставить символы в подстроке, где подстрока – участок строки

1. вторую половину строки;
2. символы, имеющие четные индексы;
3. символы первой половины строки;
4. подстроку из трех подряд слов, последним из которых является предпоследнее слово строки;
5. k-ое слово в строке;
6. подстроку из двух подряд слов, начинающейся с третьего по порядку слова;
7. первое слово в строке;
8. четвертое с конца слово, начинающееся с гласной буквы;
9. последнее слово в строке;
10. второе с конца слово, начинающееся с согласной буквы;
11. второе слово, начинающееся с гласной буквы;
12. третье слово, начинающееся с согласной буквы;
13. первую по порядку подстроку из подряд идущих цифр (предполагается, что группа из подряд идущих цифр содержит не менее двух цифр, а таких групп не менее одной);
14. символы подстроки, заключенные между третьей гласной и предпоследней согласной;
15. символы, имеющие нечетные индексы.

#### **Часть 4. Зеркальная (реверс) перестановка символов в подстроке**

**Варианты индивидуальных заданий.** Дана строка (набор символов и разделителей). Зеркально переставить символы в подстроке (локальный реверс), где подстрока – участок строки.

1. первую по порядку подстроку из подряд идущих цифр (предполагается, что группа из подряд идущих цифр содержит не менее двух цифр, а таких групп не менее одной);
2. символы подстроки, заключенные между третьей гласной и предпоследней согласной;
3. символы, имеющие нечетные индексы;
4. вторую половину строки;
5. символы, имеющие четные индексы;
6. символы первой половины строки;
7. подстроку из трех подряд слов, последним из которых является предпоследнее слово строки;
8. k-ое слово в строке (k вводится с клавиатуры);
9. подстроку из двух подряд слов, начинающейся с третьего по порядку слова;

10. первое слово в строке;
11. четвертое с конца слово, начинающееся с гласной буквы;
12. последнее слово в строке;
13. второе с конца слово, начинающееся с согласной буквы;
14. второе слово, начинающееся с гласной буквы;
15. третье слово, начинающееся с согласной буквы.

## Литература

1. Блинов, И.Н. Java from EPAM / И.Н. Блинов, В.С. Романчик. - Минск: Четыре четверти, 2020. - 560 с.
2. Язык Java. Основы синтаксиса / А.С. Кравчук, А.И. Кравчук, Е.В. Кремень [Электронный документ] URL: <https://elib.bsu.by/handle/123456789/290065> (Дата доступа: 06.12.2022)
3. Язык Java. Операторы управления программой / А.С. Кравчук, А.И. Кравчук, Е.В. Кремень [Электронный документ] URL: <https://elib.bsu.by/handle/123456789/290066> (Дата доступа: 06.12.2022)
4. Язык Java. Массивы, строки, классы обертки базовых типов / А.С. Кравчук, А.И. Кравчук, Е.В. Кремень [Электронный документ] URL: <https://elib.bsu.by/handle/123456789/290067> (Дата доступа: 06.12.2022)
5. Язык Java. Методы, стек и куча / А.С. Кравчук, А.И. Кравчук, Е.В. Кремень [Электронный документ] URL: <https://elib.bsu.by/handle/123456789/290068> (Дата доступа: 06.12.2022)
6. Кравчук, А.И. Сборник лабораторных работ и примеров решения задач по алгоритмизации и программированию на языке Си / А.И. Кравчук, А.С. Кравчук - Минск: Технопринт, 2002. - 116 с.
7. Информатика и программирование: лабораторный практикум по программированию на языке C++ : учеб.-метод. пособие / В.Ю. Наумов, О.А. Авдеюк; ВолгГТУ. – 2-е изд., испр. – Волгоград, 2018. – 136 с.
8. Игнатъева, О.В. Информатика и программирование. В 2 ч. Ч. 1 / О.В. Игнатъева; ФГБОУ ВО РГУПС. – Ростовн/Д, 2017. – 205 с.
9. Методические указания и рекомендации по выполнению лабораторных работ по дисциплине Технологии программирования для студентов специальности 5В070400 – «Вычислительная техника и программное обеспечение» - [Электронный документ] URL: <https://referatdb.ru/informatika/262454/index.html?page=14> (Дата обращения: 08.08.22)
10. Лабораторная работа № 3. Условные операторы языка С – [Электронный документ] URL: [https://mirea.thedrhaх.pw/Бакалавриат\(ИСИТ\)/2\\_семестр/Программирование/Задания/Лабораторная\\_работа\\_3.\\_Условные\\_операторы.pdf](https://mirea.thedrhaх.pw/Бакалавриат(ИСИТ)/2_семестр/Программирование/Задания/Лабораторная_работа_3._Условные_операторы.pdf) (Дата обращения: 08.08.22)

Учебное издание

**Кравчук Александр Степанович**  
**Кравчук Анжелика Ивановна**  
**Кремень Елена Васильевна**

**Введение**  
**в язык JAVA.**  
**Сборник заданий**  
**и тематических примеров**

**Учебные материалы**  
**для студентов специальности 1-31 03 08**  
**«Математика и информационные технологии**  
**(по направлениям)»**

В авторской редакции

Ответственный за выпуск *Е. В. Кремень*

Подписано в печать 20.02.2023. Формат 60×84/16. Бумага офсетная.  
Усл. печ. л. 3,26. Уч.- изд. л. 3,07. Тираж 50 экз. Заказ

Белорусский государственный университет.  
Свидетельство о государственной регистрации издателя, изготовителя,  
распространителя печатных изданий № 1/270 от 03.04.2014.  
Пр. Независимости 4, 220030, Минск.

Отпечатано с оригинал-макета заказчика  
на копировально-множительной технике  
механико-математического факультета  
Белорусского государственного университета.  
Пр. Независимости 4, 220030, Минск.