

# Java. Лабораторная работа №3

## Задание 1

### i Условие

Факториал положительного целого числа  $n$  (обозначается  $n!$  и произносится как « $n$  факториал») равен произведению всех положительных целых чисел от 1 до  $n$ . Напишите программу, которая вычисляет факториалы чисел от 1 до 20. Используйте тип `long`. Отобразите результаты в табличном формате. Какая сложность может помешать вам вычислить факториал числа 100?

```
public class Main {  
    public static void main(String[] args) {  
        long n_factorial = 1;  
  
        for (int n = 1; n <= 20; n++) {  
            n_factorial *= n;  
            System.out.print(n + ": " + n_factorial + " | ");  
        }  
    }  
}
```

100! нельзя вычислить в типе `long`, потому что число слишком большое и не вписывается в рамки этого типа данных.

## Задание 2

### i Условие

Онлайн-ритейлер продаёт пять товаров с такими розничными ценами:

- Товар 1 – \$2.98
- Товар 2 – \$4.50
- Товар 3 – \$9.98
- Товар 4 – \$4.49
- Товар 5 – \$6.87

Напишите программу, которая считывает последовательность пар чисел:

- Номер товара.
- Количество проданных единиц.

Ваша программа должна использовать оператор `switch` для определения итоговой цены для каждого товара. Она должна вычислять и отображать общую стоимость всех проданных товаров. Используйте цикл с сентинелем (специальным значением, сигнализирующим об окончании ввода), чтобы определить, когда программа должна прекратить выполнение цикла и вывести итоговые результаты.

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        double total_sum = 0.0;

        while (true) {
            System.out.print("Введите номер товара (1-5, 0 - конец ввода):");
        };
        int product_number = scanner.nextInt();

        if (product_number == 0) {
            break;
        }

        System.out.print("Введите количество проданных единиц: ");
        int quantity = scanner.nextInt();

        double price;

        switch (product_number) {
            case 1:
                price = 2.98;
                break;
            case 2:
                price = 4.50;
                break;
            case 3:
                price = 1.75;
                break;
            case 4:
                price = 1.50;
                break;
            case 5:
                price = 7.99;
                break;
        }

        total_sum += price * quantity;
    }
}
```

```

        price = 9.98;
        break;
    case 4:
        price = 4.49;
        break;
    case 5:
        price = 6.87;
        break;
    default:
        System.out.println("Неверный номер товара, строка
пропущена.");
        continue;
    }

    double cost = price * quantity;
    total_sum += cost;

    System.out.println("Стоимость по этому товару: " + cost);
}

System.out.println("Общая стоимость всех проданных товаров: " +
total_sum);

scanner.close();
}
}

```

## Задание 3

### Условие

Вычислите значение числа  $\pi$ , используя бесконечный ряд:

$$\pi = \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7}$$

Выведите таблицу, показывающую приближённое значение числа  $\pi$  при вычислении первых 200000 членов этого ряда. Определите сколько членов необходимо использовать, прежде чем впервые получится значение, начинающееся с 3.14159?

```

public class Main {
    public static void main(String[] args) {
        final int max_terms = 200000; // максимум членов ряда
        double sum = 0.0;
    }
}

```

```
double sign = 1.0;

double target_pi = 3.14159;
int terms_for_target = -1;

System.out.println("Число членов      Приближённое π");
System.out.println("");

for (int n = 1; n <= max_terms; n++) {
    double denominator = 2.0 * n - 1.0;
    double term = sign * (1.0 / denominator);

    sum = sum + term;
    sign = -sign; // меняем знак на противоположный

    double pi_approx = 4.0 * sum;

    System.out.println(n + "           " + pi_approx);

    // первая точка, где π ≈ 3.14159 (до 5 знаков после запятой)
    if (terms_for_target == -1) {
        double rounded = Math.round(pi_approx * 100000.0) /
100000.0;
        if (rounded == target_pi) {
            terms_for_target = n;
        }
    }
}

System.out.println();
if (terms_for_target != -1) {
    System.out.println("Впервые значение 3.14159 получается после "
+ terms_for_target + " членов ряда.");
} else {
    System.out.println("За " + max_terms
+ " членов ряда значение 3.14159 не было достигнуто.");
}
}
```