Java: знакомство и как пользоваться базовым API (семинары)

Задание 1. Нахождение факториала числа

Напишите метод `factorial`, который принимает число `n` и возвращает его факториал. Если число `n < 0`, метод должен вернуть `-1`.

```
class Answer {

    public int factorial(int n) {

        // Введите свое решение ниже

        ...

    }

}


// Не удаляйте этот класс - он нужен для вывода результатов на экран и проверки

public class Printer {

    public static void main(String[] args) {

        int n = 5;


        if (args.length > 0) {

            n = Integer.parseInt(args[0]);

        }


        // Вывод результата на экран

        Answer ans = new Answer();

        int itresume_res = ans.factorial(n);
```

```
        System.out.println(itresume_res);

    }

}
```

Подсказка № 1

Факториал числа nnn обозначается как n!n!n! и вычисляется путем умножения всех целых чисел от 1 до n. Например, 5!=5×4×3×2×1=120. Помните, что факториал для 0 всегда равен 1, то есть 0!=1.

Подсказка № 2

Чтобы вычислить факториал числа n, вам нужно использовать цикл для последовательного умножения чисел от 1 до n. Если n отрицательное, метод должен сразу возвращать -1, так как факториал отрицательного числа не определен.

Подсказка № 3

Начните с проверки, является ли n отрицательным. Если это так, верните -1. Если нет, используйте цикл `for` для умножения всех чисел от 1 до n и возвращения результата. Для хранения результата используйте переменную, которую вы будете обновлять на каждом шаге цикла.

Эталонное решение:

```
class Answer {

   public int factorial(int n) {

        // Введите свое решение ниже

        if (n < 0) {

            return -1;

        }

        int result = 1;

        for (int i = 2; i <= n; i++) {

            result *= i;

        }

        return result;
```

```
    }
}


// Не удаляйте этот класс - он нужен для вывода результатов на экран и проверки
public class Printer {
    public static void main(String[] args) {
        int n = 5;

        if (args.length > 0) {
            n = Integer.parseInt(args[0]);
        }

        // Вывод результата на экран
        Answer ans = new Answer();
        int itresume_res = ans.factorial(n);
        System.out.println(itresume_res);
    }
}
```

Задача 2. Вывод всех четных чисел от 1 до 100

Напишите метод `printEvenNums`, который выведет на экран все четные числа в промежутке от 1 до 100, каждое на новой строке.

```
class Answer {
    public static void printEvenNums() {
        // Напишите свое решение ниже
        ...
```

```
    }
}


// Не удаляйте этот класс - он нужен для вывода результатов на экран и проверки
public class Printer {
    public static void main(String[] args) {
        Answer ans = new Answer();
        ans.printEvenNums();
    }
}
```

Подсказка № 1

Для проверки, является ли число четным, вам нужно использовать оператор остатка от деления `%`. Четное число — это число, которое делится на 2 без остатка. Например, если `n % 2 == 0`, то `n` является четным.

Подсказка № 2

Для перебора всех чисел в промежутке от 1 до 100, используйте цикл `for`. Вы можете задать диапазон от 1 до 100 включительно и проверить каждое число на четность внутри цикла.

Подсказка № 3

Внутри цикла, если число четное, используйте метод `System.out.println()` для вывода его на экран. Каждое число должно выводиться на новой строке.

Эталонное решение:

```
class Answer {
    public static void printEvenNums() {
        // Напишите свое решение ниже
```

```
        for (int i = 2; i <= 100; i += 2) {

            System.out.println(i);

        }

    }

}


// Не удаляйте этот класс - он нужен для вывода результатов на экран
и проверки

public class Printer {

    public static void main(String[] args) {

        Answer ans = new Answer();

        ans.printEvenNums();

    }

}
```

Задача 3. Подсчет суммы цифр числа

Напишите метод `sumDigits`, который принимает целое число `n` и возвращает сумму его цифр.

```
class Answer {

    public int sumDigits(int n) {

        // Введите свое решение ниже

        ...

    }

}


// Не удаляйте этот класс - он нужен для вывода результатов на
экран и проверки
```

```java
public class Printer {

    public static void main(String[] args) {

        int n = 12345;


        if (args.length > 0) {

            n = Integer.parseInt(args[0]);

        }


        // Вывод результата на экран

        Answer ans = new Answer();

        int itresume_res = ans.sumDigits(n);

        System.out.println(itresume_res);

    }

}
```

Подсказка № 1

Чтобы получить последнюю цифру числа, используйте оператор остатка от деления (`%`). Например, `n % 10` вернет последнюю цифру числа `n`.

Подсказка № 2

После того как вы получили последнюю цифру числа, вам нужно удалить эту цифру из числа. Это можно сделать с помощью целочисленного деления (`/`). Например, `n = n / 10` удалит последнюю цифру из числа `n`.

Подсказка № 3

Используйте цикл `while` для того, чтобы повторять операции с числами, пока само число не станет равно 0. В каждой итерации добавляйте последнюю цифру числа к сумме.

Эталонное решение:

```java
class Answer {

   public int sumDigits(int n) {

       // Введите свое решение ниже

       int sum = 0;

       while (n != 0) {

           sum += n % 10;

           n /= 10;

       }

       return sum;

   }

}



// Не удаляйте этот класс - он нужен для вывода результатов на экран и проверки

public class Printer {

   public static void main(String[] args) {

       int n = 12345;



       if (args.length > 0) {

           n = Integer.parseInt(args[0]);

       }



       // Вывод результата на экран

       Answer ans = new Answer();

       int itresume_res = ans.sumDigits(n);

       System.out.println(itresume_res);

   }
```

```
}
```

Задача 4*. Нахождение максимального из трех чисел

Реализуйте две функции:

1. Функция `findMaxOfTwo` должна принимать два числа и возвращать максимальное из них, используя только знак сравнения.
2. Функция `findMaxOfThree` должна принимать три числа и находить максимальное из них, используя первую функцию.

```
class Answer {

    // Функция для нахождения максимума из двух чисел
    public int findMaxOfTwo(int a, int b) {
        ...
    }


    // Функция для нахождения максимума из трех чисел
    public int findMaxOfThree(int a, int b, int c) {
        ...
    }
}


// Не удаляйте этот класс - он нужен для вывода результатов на экран и проверки
public class Printer {
    public static void main(String[] args) {
        int a = 5, b = 10, c = 3;
```

```
        if (args.length == 3) {
            a = Integer.parseInt(args[0]);
            b = Integer.parseInt(args[1]);
            c = Integer.parseInt(args[2]);
        }

        // Вывод результата на экран
        Answer ans = new Answer();
        int itresume_res = ans.findMaxOfThree(a, b, c);
        System.out.println(itresume_res);
    }
}
```

Подсказка № 1

Для функции `findMaxOfTwo` вам нужно сравнить два числа `a` и `b`. Используйте оператор сравнения (`>`) для того, чтобы определить, какое из двух чисел больше. Например, если `a` больше `b`, функция должна вернуть `a`, иначе — `b`.

Подсказка № 2

В Java можно использовать тернарный оператор (`? :`), чтобы компактно записать условие сравнения. Тернарный оператор работает как сокращенная форма записи `if-else` и отлично подходит для возврата максимального значения между двумя числами.

Подсказка № 3

Для функции `findMaxOfThree` сначала найдите максимальное значение между первыми двумя числами (`a` и `b`), используя функцию `findMaxOfTwo`. Затем сравните результат с третьим числом (`c`) и снова используйте функцию `findMaxOfTwo`, чтобы найти максимальное значение из всех трех чисел.

Подсказка № 4

После того как вы написали функции, протестируйте их на нескольких примерах. Например, проверьте такие наборы чисел: (5, 10, 3), (12, 4, 9), (7, 7, 7). Убедитесь, что ваша программа правильно находит максимальное число в каждом случае.

Эталонное решение:

```
class Answer {

    // Функция для нахождения максимума из двух чисел

    public int findMaxOfTwo(int a, int b) {

        return (a > b) ? a : b;

    }



    // Функция для нахождения максимума из трех чисел

    public int findMaxOfThree(int a, int b, int c) {

        // Сначала находим максимум между a и b, а затем сравниваем
его с c

        return findMaxOfTwo(findMaxOfTwo(a, b), c);

    }

}



// Не удаляйте этот класс - он нужен для вывода результатов на экран
и проверки

public class Printer {

    public static void main(String[] args) {

        int a = 5, b = 10, c = 3;



        if (args.length == 3) {

            a = Integer.parseInt(args[0]);

            b = Integer.parseInt(args[1]);
```

```
            c = Integer.parseInt(args[2]);

        }



        // Вывод результата на экран

        Answer ans = new Answer();

        int itresume_res = ans.findMaxOfThree(a, b, c);

        System.out.println(itresume_res);

    }

}
```