## День 2

Темы:

[Урок 6: Условный оператор if.](https://www.youtube.com/watch?v=ryR033ld_N0&list=PLAma_mKffTOSUkXp26rgdnC0PicnmnDak&index=6&ab_channel=alishev)

[Урок 7: Ввод данных. Класс Scanner.](https://www.youtube.com/watch?v=Y2iB_DwdyfM&list=PLAma_mKffTOSUkXp26rgdnC0PicnmnDak&index=7&ab_channel=alishev)

Доп. статьи:

1. Статья про арифметические операторы в Java. Они необходимы для решения задач этого и последующих дней.

<https://metanit.com/java/tutorial/2.3.php>

Доп. информация про логические операторы И, ИЛИ, НЕ

В Java есть следующие операторы:  
&& - логическое “И”

|| - логическое “ИЛИ”

! - логическое “НЕ”  
  
Эти операторы работают с булевскими значениями - true (истина) и false (ложь). Как и известные нам математические операторы +, -, /, \*, логические операторы принимают на вход значения (операнды) и возвращают значение - результат операции. Только если в случае с математическими операторами, входные и возвращаемые значения - это числа, в случае с логическими операторами - входные и возвращаемые значения - это булевские значения (true либо false).

Логический оператор && (И) возвращает true только в том случае, если оба входных значения равны true. Во всех остальных случаях этот оператор возвращает false. Легко запомнить: оператор логического “И” требует, чтобы первое И второе значение были true, чтобы результат был true.

**Пример кода:**

**boolean** a = **true**;

**boolean** b = **true**;

**boolean** result = a && b; // a,b - операнды, && - операция логического “И”

System.out.println(result); // true

**Другой пример:**

**boolean** a = **true**;

**boolean** b = **false**;

**boolean** result = a && b;

System.out.println(result); // false

Логический оператор || (ИЛИ) возвращает true в том случае, если хотя бы одно из входных значения равно true. Если ни одно из входных значений не равно true, оператор ИЛИ возвращает false. Легко запомнить: оператор логического “ИЛИ” требует, чтобы первое ИЛИ второе значение были true, чтобы результат был true.

**Пример кода:**

**boolean** a = **false**;

**boolean** b = **true**;

**boolean** result = a || b;

System.out.println(result); // true

**Другой пример:**

**boolean** a = **false**;

**boolean** b = **false**;

**boolean** result = a || b;

System.out.println(result); // false

Логический оператор ! (НЕ) принимает на вход только одно булевское значение и инвертирует его. Из true делается false, а из false делается true. Этот логический оператор еще называют оператором логического отрицания.

**Пример кода:**

**boolean** a = **false**;

**boolean** b = **true**;

System.out.println(!a); // true

System.out.println(!b); // false

Вышеописанные логические операторы особенно полезны в условном операторе if, так как они позволяют нам комбинировать несколько условий.

Допустим, мы хотим, чтобы код в блоке if выполнялся в том случае, если значение, лежащее в переменной a больше, чем 5, но меньше, чем 10. Код для этого будет выглядеть следующим образом:

**if** (a > **5** && a < **10**) {

// код здесь

}

**Задачи:**

1. Реализовать программу, которая принимает на вход через консоль с помощью класса Scanner, число, соответствующее количеству этажей в здании. Используя условный оператор if, необходимо вывести в консоль сообщение о типе такого дома.

Условия: если этажей 1-4 - “Малоэтажный дом”, 5-8 - “Среднеэтажный дом”, 9 и более - “Многоэтажный дом”. Также, необходимо учесть что может быть введено отрицательное значение, в таком случае сообщить “Ошибка ввода”.

2. Есть два числа, которые задаются пользователем через консоль (назовем эти два числа a и b). Используя цикл for, вывести все числа из диапазона между a и b, которые делятся на 5 без остатка, но при этом **не** делятся на 10 без остатка. Например, число 15 подходит под наше условие (делится на 5 без остатка и не делится на 10 без остатка), но число 20 не подходит под наше условие (делится на 5 без остатка и делится на 10 без остатка). Сами числа a и b в диапазоне не учитывать.

Если a >= b вывести сообщение "Некорректный ввод".

**Пример:**

Вводим в консоли: 7 78

Вывод: 15 25 35 45 55 65 75

\*Для получения из консоли двух чисел, разделенных пробелом, можно использовать метод nextInt() вызвав его дважды.

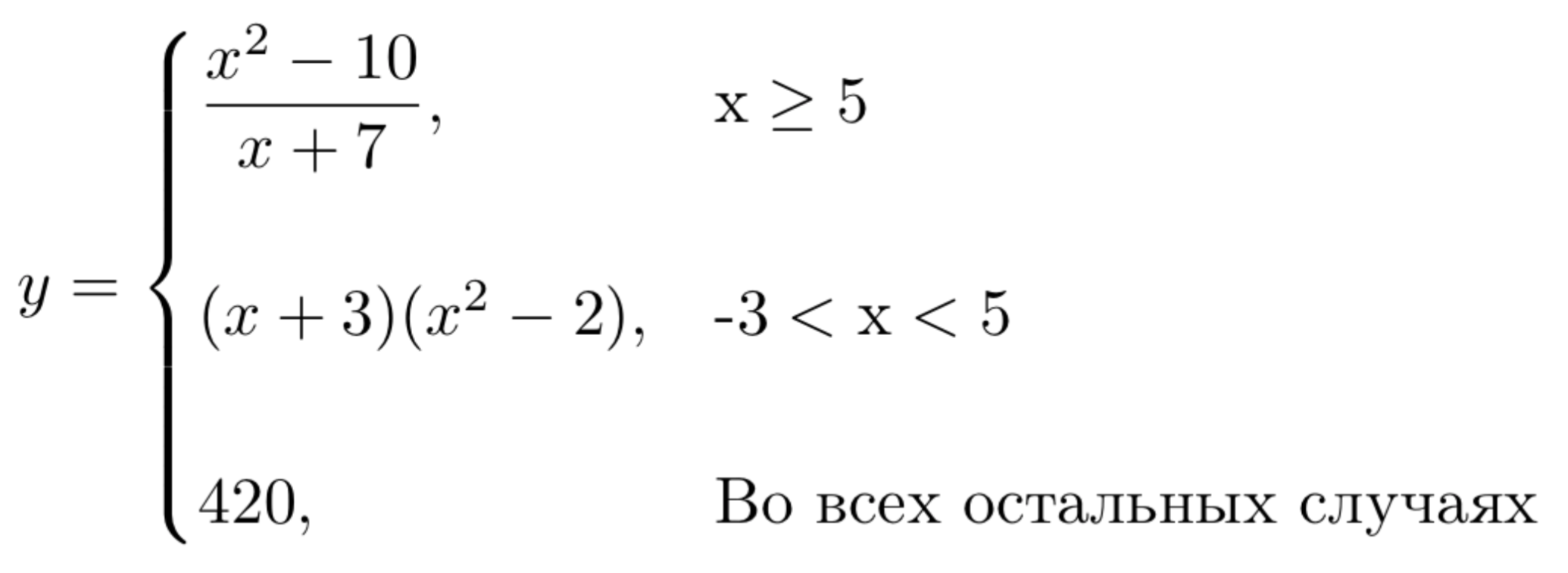
Scanner scanner = **new** Scanner(System.in);

**int** a = scanner.nextInt();

**int** b = scanner.nextInt();

3. Реализовать программу №2, используя цикл while.

4.



Реализовать программу, которая принимает на вход через консоль с помощью класса Scanner, число x. Для этого числа, по формуле выше, необходимо вычислить значение y и вывести его в консоль, используя метод System.out.println().

(Для этих вещественных чисел x и y необходимо использовать тип double и метод nextDouble() у Scanner’а соответственно, чтобы считать из консоли число x).

\*Пояснение.

Формула для вычисления значения y содержит две составляющие - набор выражений для вычисления значения y и условие, при котором то или иное выражение “срабатывает”. Выполнение того или иного условия зависит от значения x. Например, если x = 100, будет использовано самое первое выражение. С помощью него, используя значение x будет вычислено значение y.

y = (100^2 - 10) / (100 + 7) = 93.3644859813