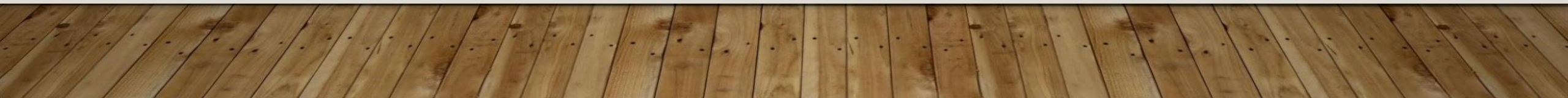
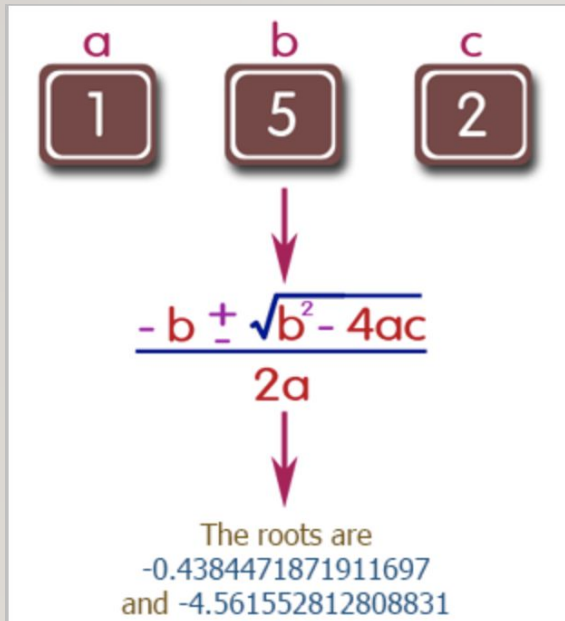


ЗАНЯТИЕ 5



IF-ELSE ПОВТОРЕНИЕ

- I) Решить квадратное уравнение $ax^2 + bx + c = 0$ через if-else
- Входные параметры: a, b, c.



Если $D > 0$,
то 2 корня

$$x_{1,2} = \frac{-b \pm \sqrt{D}}{2a}$$

Если $D = 0$,
то 1 корень

$$x = \frac{-b}{2a}$$

Если $D < 0$,
то корней нет

P.S 1: возвести число во 2-ую
степени – умножить на себя
либо использовать

Math.pow(number, powerToUse)

P.S 2: корень – степень 0.5

$$ax^2 + bx + c = 0$$

$$x = \frac{-b \pm \sqrt{D}}{2a}, \text{ где } D = b^2 - 4ac$$

FOR LOOP РАЗМИНКА

- 1) Напечатать числа от 1 до 10
- 2) Найти факториал любого числа, введенного с клавиатуры
- 3) Найти сумму $1 + 1/2 + \dots + 1/n$

WHILE

Цикл while итерируется в блоке кода, пока условие true:

Syntax

```
while (condition) {  
    // code block to be executed  
}
```

3 элемента цикла:

- 1) Ключевое слово while
- 2) Предикат. Условие, которое указывается в скобках после while. Это условие вычисляется на каждой итерации.
- 3) Тело цикла. Блок кода в фигурных скобках.

WHILE

В приведенном ниже примере код в цикле будет выполняться снова и снова, пока переменная (i) меньше 5:

```
int i = 0;
while (i < 5) {
    System.out.println(i);
    i++;
}
```

ATTENTION: Не забудьте увеличить используемую в условии переменную, иначе цикл никогда не закончится!

ЗАДАНИЕ НА WHILE

- 1) Написать программу, которая выполняет обратный отсчет от 10 до 0
- 2) Найти сумму $1 + 1/2 + \dots + 1/n$

WHILE VS FOR

while

```
int i = 3;
while (i >= 0)
{
    System.out.println(i);
    i--;
}
```

```
int i = 0;
while (i < 3)
{
    System.out.println(i);
    i++;
}
```

```
while (true)
    System.out.println("e");
```

for

```
for (int i = 3; i >= 0; i--)
{
    System.out.println(i);
}
```

```
for (int i = 0; i < 3; i++)
{
    System.out.println(i);
}
```

```
for (; true; )
    System.out.println("e");
```

DO-WHILE

```
do  
{  
    //Операторы  
} while(логическое выражение);
```

Обратите внимание, что логическое выражение появляется в конце цикла, так что операторы в цикле выполняются один раз, прежде чем пройдут проверку на логическое условие.

Если логическое выражение истинно, контроль переходит обратно, чтобы выполнить операторы, и они в цикле выполняются снова. Этот процесс повторяется до тех пор, пока логическое выражение не станет ложным.

DO-WHILE

```
int x = 10;

do{
    System.out.print("Значение x: " + x );
    x++;
    System.out.print("\n");
} while( x < 15 );
```

А если написать $x < 10$?

BREAK

- Оператор **break** позволяет выйти из цикла в любой его момент, даже если цикл не закончил свою работу:

```
for (int i = 0; i < 10; i++){  
    if (i == 5)  
        break;  
    System.out.println(i);  
}
```

Когда счетчик станет равным 5, сработает оператор break, и цикл завершится.

CONTINUE

- Теперь сделаем так, чтобы если число равно 5, цикл не завершался, а просто переходил к следующей итерации. Для этого используем оператор **continue**:

```
for (int i = 0; i < 10; i++){  
    if (i == 5)  
        continue;  
    System.out.println(i);  
}
```

В этом случае, когда выполнение цикла дойдет до числа 5, программа просто пропустит это число и перейдет к следующему.

RETURN

- Выход полностью

```
for (int i = 0; i < 10; i++) {  
    int y = 11;  
    if (i == 5) {  
        return;  
    }  
  
    y++;  
    System.out.println("y = " + y);  
  
    System.out.println("i = " + i);  
}  
  
System.out.println("After cycle");
```

ПРАКТИКА (+ДЗ)

<https://metanit.com/java/tutorial/2.6.php>

- 1) Вычислить: $1+2+4+8+\dots+256$
- 2) Напишите программу вывода всех четных чисел от 2 до 100 включительно
- 3) Напишите программу, определяющую сумму всех нечетных чисел от 1 до 99
- 4) Даны числа от 0 до 100.
 - Посчитать сумму четных и вывести на экран
 - Посчитать сумму нечетных и вывести на экран
 - Найти общую сумму всех чисел