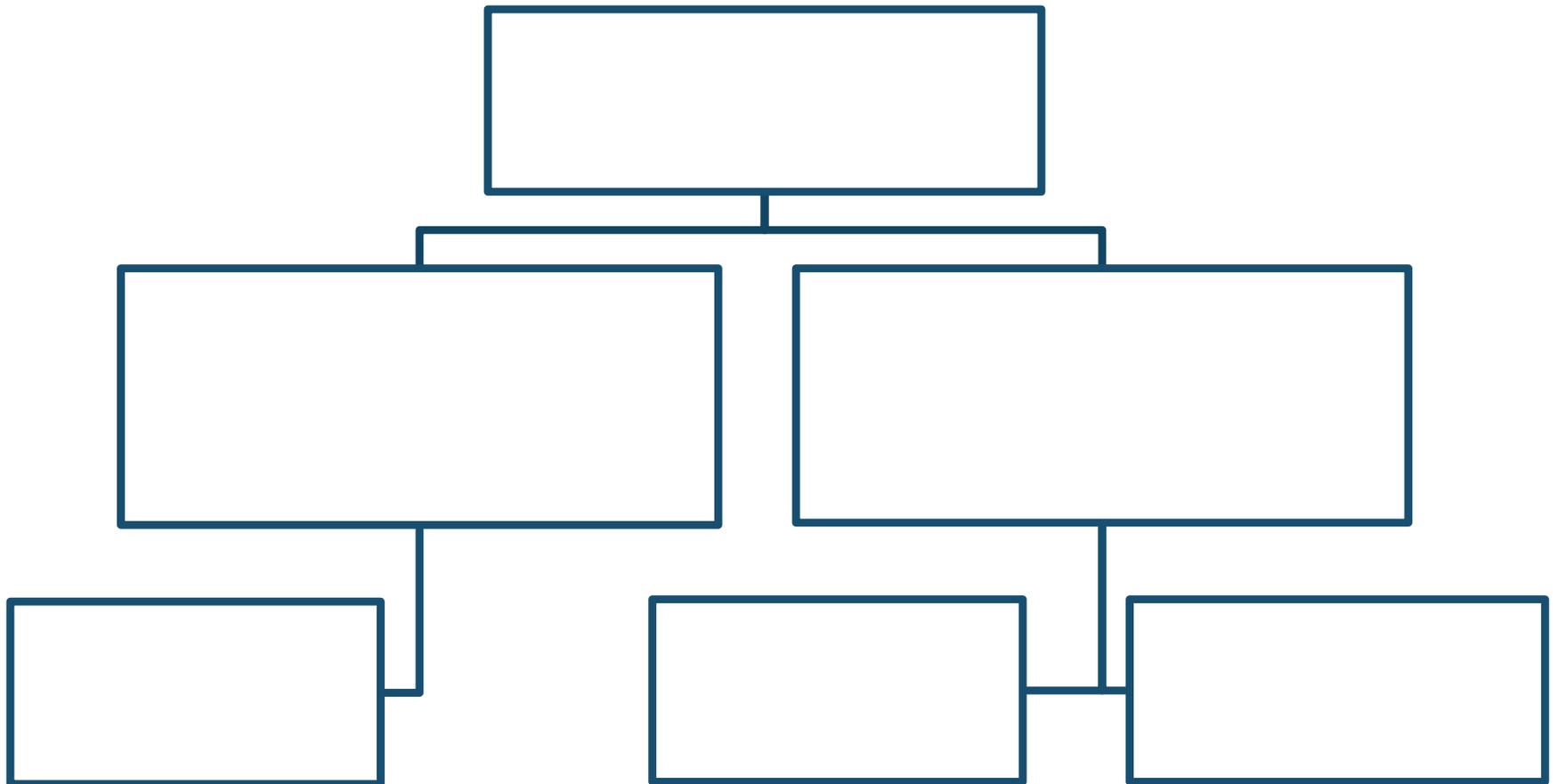


**Java**

Циклы

# Циклы

**Цикл** – это многократное выполнение одинаковой последовательности действий.



# Цикл с переменной

```
for (инициализация; условие ; итерация)
{
// тело цикла
}
```

**Инициализация** — первый параметр, который содержит переменную и ее начальное значение. С помощью этой переменной будет подсчитываться количество повторений цикла.

**Условие** — второй параметр, содержит некоторое логическое выражение — условие при котором будет выполняться цикл.

**Итерация** — третий параметр, выражение, изменяющее переменную после каждого шага цикла.

## Примеры:

```
for (int i = 1; i <= 10; i++) {  
    System.out.print(i + " ");  
}
```

Представленная программа выводит на экран числа от 10 до -10:

```
for (int s = 10; s > -11; s--) {  
    System.out.print(s + " ");  
}
```

Представленная программа выводит на экран нечётные числа от 1 до 33:

```
for (int i = 1; i <= 33; i += 2) {  
    System.out.print(i + " ");  
}
```

## Примеры:

Представленная программа вычислит сумму элементов фрагмента последовательности 2, 4, 6, 8,... 98, 100:

```
int sum=0;
for (int j = 2; j <= 100; j+=2) {
    sum += j;
}
System.out.println(sum);
```

Представленная программа будет возводить число a в натуральную степень из переменной n:

```
double a = 2; int n = 10;
double res = 1;
for (int i = 1; i <= n; i++) {
    res *= a;
}
System.out.println(res);
```

# Свойства цикла for:

- *условие* проверяется в начале очередного шага цикла, если оно ложно цикл не выполняется;
- *изменения* (третья часть в заголовке) выполняются в конце очередного шага цикла;
- если *условие* никогда не станет ложным, цикл может продолжаться бесконечно (зацикливание)

```
for (i=1; i<8; i++) { i--; }
```



Не рекомендуется менять переменную цикла в теле цикла!

- если в теле цикла один оператор, скобки `{}` можно не ставить:

```
for (i = 1; i < 8; i++) a += b;
```

# Чему равно значение переменной a?

```
a = 1;  
for (i=1; i<4; i++) a++;
```

a = 4

```
a = 1;  
for (i=1; i<4; i++) a = a+i;
```

a = 7

```
a = 1; b=2;  
for (i=3; i >= 1; i--) a += b;
```

a = 7

```
a = 1;  
for (i=1; i >= 3; i--) a++;
```

a = 1

```
a = 1;  
for (i=1; i <= 4; i--) a++;
```

зацикливание

## Примеры:

Представленная программа выведет на экран такую последовательность «0 -1 -4 -9 -16 -25:

```
for (int a = 0, b = 0; a-b <= 10; a++, b--) {  
    System.out.print(a * b + " ");  
}
```

В результате работы следующего примера на экран будут выведены только числа «1 2 3 4 Конец:

```
for (int a=1; a<=10; a++) {  
    if(a == 5) {  
        break;  
    }  
    System.out.print(a + " ");  
}  
System.out.print("Конец");
```

# Вложенные циклы

# Псевдографика

```
for (int i = 0; i < 10; i++) {  
    for (int j = 0; j < 10; j++) {  
        System.out.print("*");  
    }  
    System.out.println();  
}
```

```
for (int i = 0; i < 10; i++) {  
    for (int j = 0; j <= i; j++) {  
        System.out.print("*");  
    }  
    System.out.println();  
}
```

# Псевдографика

```
for (int i = 10; i > 0; i--) {  
    for (int j = 0; j < i; j++) {  
        System.out.print( "*" );  
    }  
    System.out.println();  
}
```

```
for (int i = -10; i < 10; i++) {  
    for (int j = 0; j <= i*i; j++) {  
        System.out.print( "*" );  
    }  
    System.out.println();  
}
```

# Циклы с условием

Цикл `while`

# Цикл с условием

```
while ( условие ) {  
    // тело цикла  
}
```

## Особенности:

- МОЖНО ИСПОЛЬЗОВАТЬ СЛОЖНЫЕ УСЛОВИЯ:

```
while ( a < b && b < c ) { ... }
```

- если в теле цикла только один оператор, скобки **{ }** можно не писать:

```
while ( a < b ) a ++;
```

# Свойства цикла while

- условие пересчитывается **каждый раз** при входе в цикл
- если условие на входе в цикл ложно, цикл не выполняется ни разу

```
a = 4; b = 6;  
while ( a > b ) a -= b;
```

- если условие никогда не станет ложным, программа **зацикливается**

```
a = 4; b = 6;  
while ( a < b ) d = a + b;
```

# Сколько раз выполняется цикл?

```
a = 4; b = 6;  
while ( a < b ) a ++;
```

2 раза

a = 6

```
a = 4; b = 6;  
while ( a < b ) a += b;
```

1 раз

a = 10

```
a = 4; b = 6;  
while ( a > b ) a ++;
```

0 раз

a = 4

```
a = 4; b = 6;  
while ( a < b ) b = a - b;
```

1 раз

b = -2

```
a = 4; b = 6;  
while ( a < b ) a --;
```

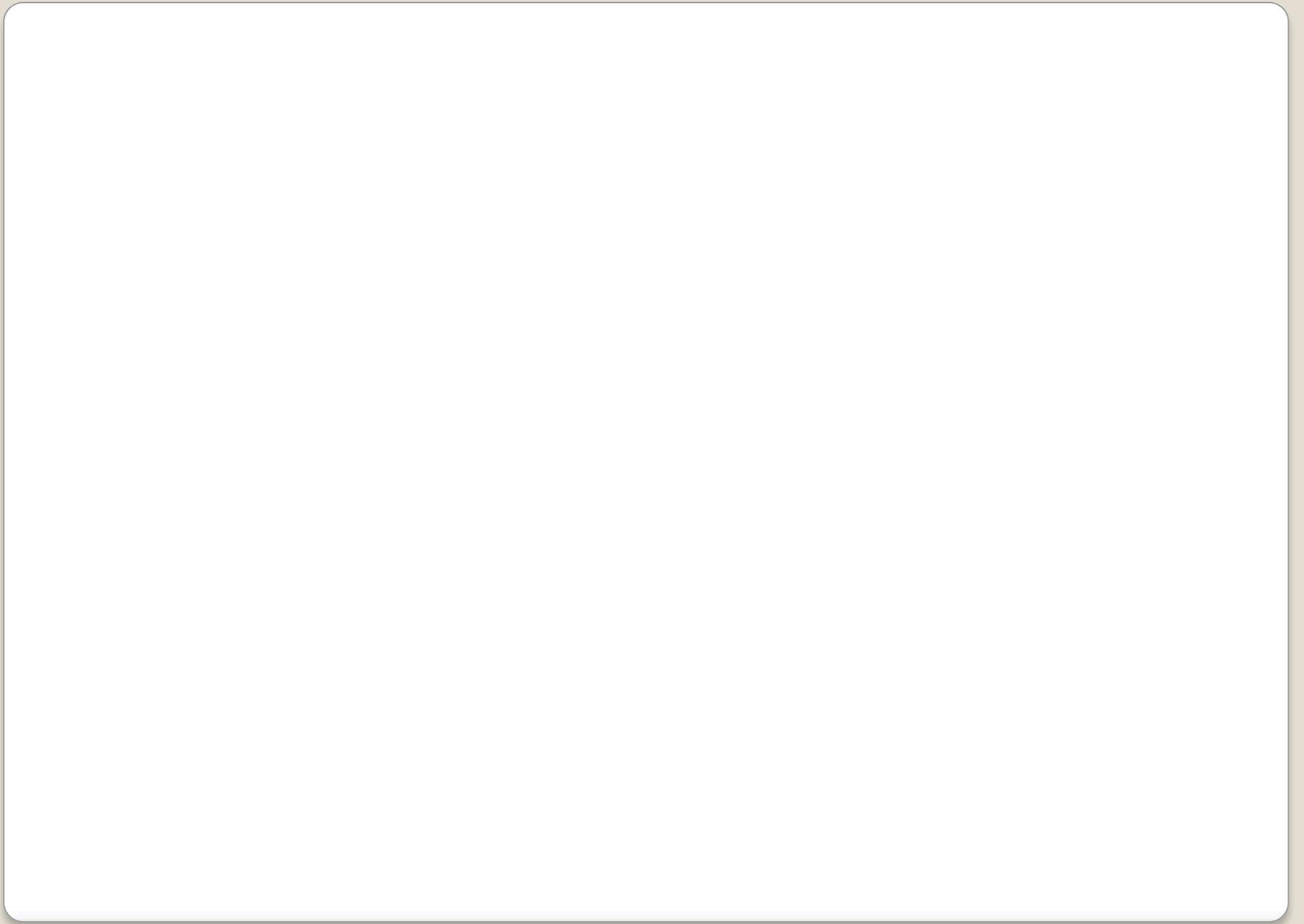
зацикливание

## Дано число x напечатать его цифры

```
while (x!=0) {  
    System.out.println(x % 10);  
    x/= 10;  
}
```

Дано целое число x. Найти количество и сумму его цифр.

```
int s=0, k=0;  
while (x!=0) {  
    s += x % 10;  
    x /= 10;  
    k++;  
}  
System.out.println(s + " " + k);
```



# Последовательности

**Задача:** найти сумму всех элементов последовательности,

$$1, -\frac{1}{2}, \frac{2}{4}, -\frac{3}{8}, \frac{4}{16}, -\frac{5}{32}, \dots$$

которые по модулю больше 0,001:

$$S = 1 - \frac{1}{2} + \frac{2}{4} - \frac{3}{8} + \frac{4}{16} - \frac{5}{32} + \dots$$

**Элемент последовательности (начиная с №2):**

$$a = z \frac{b}{c}$$

<b>n</b>	1	2	3	4	5	...
<b>b</b>	1	2	3	4	5	...
<b>c</b>	2	4	8	16	32	...
<b>z</b>	-1	1	-1	1	-1	...

$$b = b + 1;$$

$$c = 2 * c;$$

$$z = -z;$$

# Программа

```
int b, c, z;
```

```
float S, a;
```

```
S = 0; z = -1;
```

```
b = 1; c = 1;
```

чтобы не было  
округления при  
делении

ые  
значения

```
while (abs(a) > 0.001) {
```

```
    S += a;
```

увеличение

```
    a = z * b / c;
```

СУММА

```
    z = -z;
```

расчет элемента  
последовательности

```
    b++;
```

```
    c *= 2;
```

```
}
```

```
printf ("S = %10.3f", S);
```

```
}
```

# Циклы с условием

Цикл `do ... while`

# Цикл с условием

```
do {  
    // тело цикла  
} while ( условие );
```

## Свойства цикла `do { ... } while`

- Тело цикла выполняется перед проверкой условия;
- Если условие на выходе из цикла ложно, цикл выполнится один раз
- После слова **while** («пока...») ставится условие продолжения цикла.

# Примеры

```
int i =10;  
do{  
    System.out.println("i= " + i);    i++;  
} while (i < 5);
```

```
int summa = 0, number;  
Scanner in = new Scanner(System.in);  
do {  
    number = in.nextInt();  
    summa += number;  
} while (number != 0);  
System.out.println(summa);  
in.close();
```

# Сколько раз выполняется цикл?

```
a = 4; b = 6;  
do { a++; } while (a <= b);
```

3 раза  
a = 7

```
a = 4; b = 6;  
do { a += b; } while ( a <= b );
```

1 раз  
a = 10

```
a = 4; b = 6;  
do { a += b; } while ( a >= b );
```

защелкивание

```
a = 4; b = 6;  
do { b = a - b; } while ( a >= b );
```

2 раза  
b = 6

```
a = 4; b = 6;  
do { a += 2; } while ( a >= b );
```

защелкивание