УСЛОВНЫЕ ОПЕРАТОРЫ

ТИП BOOLEAN

Если нужно представить величину из двух вариантов, используют boolean (true или false) <<*Приведите примеры*>>

Переменную именуют, начиная с is… или has. Например,
isEmpty // флаг того, что коллекция пуста
isFirstTime // флаг того, что что-то произошло впервые
hasCompleted //флаг того, что процесс завершён

Объявление:

```
1 boolean isClosed = false;
2 int a = 1; int b = 100;
boolean isBigger = a > b; //чему равно isBigger?
3 boolean hasFinished = isClosed || isBigger;
```

ЛОГИЧЕСКИЕ ОПЕРАЦИИ

Оператор Java	Имя	Тип	Краткое описание	Пример
!	Логическое “не” (отрицание)	Унарный	!x означает “не x”. Возвращает true если операнд является false. Возвращает false если операнд является true.	boolean x = true; Тогда // !x == false

&	Логическое И (AND, умножение)	Бинарный	Возвращает true если оба операнда равны true.	a = true;
				b = false;
				тогда
				a & b == false

|	Логическое ИЛИ (OR, сложение)	Бинарный	Возвращает true если хотя бы один из операндов равен true.	a = true;
				b = false;
				тогда
				a | b == true

&&	Условное И (сокращённое логическое И)	Бинарный	То же самое, что и &, но если операнд, находящийся слева от & является false, данный оператор возвращает false без проверки второго операнда.	
||	Условное ИЛИ (сокращённое логическое ИЛИ)	Бинарный	То же самое, что и |, но если оператор слева является true, оператор возвращает true без проверки второго операнда.	

^	Логическое исключающее ИЛИ (XOR)	Бинарный	Возвращает true, если один и только один из операндов равен true. Возвращает false, если оба операнда равны true или false. По сути, возвращает true, если операнды — разные.	a = true;
				b = false;
				тогда
				a ^ b == true

ОПЕРАТОР УСЛОВНОГО ПЕРЕХОДА

```
//Программа что-то делает
if (условие) {
   //если оказались здесь, то условие==true
   действия программы;
}
//оказались здесь, значит, блок if уже выполнился
(условие==true) или был пропущен (условие==false)

дальнейшие действия программы;
```

ВЕТКА ELSE

```
//Программа что-то делает
if (условие) {
   //если оказались здесь, то условие==true
   действия программы для случая true;
}else {
   //если оказались здесь, то условие==false
    действия программы для случая false;
}

//оказались здесь, значит, выполнилась одна из веток блока if

дальнейшие действия программы;
```

ПОСЛЕДОВАТЕЛЬНЫЕ IF

```
//Программа что-то делает
if (условие1) {
   действия1 программы;
}
if (условие2) {
   действия2 программы;
}
if (условие3) {
   действия3 программы;
}
//Программа что-то делает
```

ВЛОЖЕННЫЕ IF

```
//Программа что-то делает
if (условие1) {
   действия1 программы;
   if (условие2) {
      действия2 программы;
   }
} else {
   if (условие3) {
      действия3 программы;
   }
   действия4 программы;
}
```

ЦЕПОЧКА ОПЕРАТОРОВ IF-ELSE-IF

```
//Программа что-то делает
if (condition) //если оператор один, его можно не брать в {}
    statement; //но так лучше не делать!!! Всегда указывайте
{}
else if (condition)
    statement;
else if (condition)
    statement;
.
.
.
else
    statement;
//Программа что-то делает
```

ТЕРНАРНЫЙ ОПЕРАТОР (ЭЛВИСА)

Заменяет if-then-else, когда нужно вернуть значение, например, чтобы присвоить его переменной.

Переменная п = логическое_условие ? выражение1 : выражение2;

Пример (найти максимальное число из двух):
int a = 7; int b = 20;
int maxOfTwo = a > b ? a : b;