

# Условные конструкции в Java

---

**Условные операторы** используются, когда в зависимости от условия необходимо выполнить разные действия.

### Оператор if

если указанное **условие** истинно (true), выполняет **код** в фигурных скобках,  
Синтаксис:

```
if (условие) {  
    код, который выполнится, если условие истинно  
}
```

Например,

```
if (answer == 13) { // если значение переменной answer будет равно 13  
    // в консоль будет выведено: Ответ верный  
    System.out.println(“Ответ верный”);  
}
```

## Необязательный блок else

выполняется, если **условие** ложно (false)

Синтаксис:

```
if (условие) {  
    код, который выполнится, если условие истинно  
}  
else {  
    код, который выполнится, если условие ложно  
}
```

Например,

```
if (answer == 13) { // если значение переменной answer будет равно 13  
    // в консоль будет выведено: Ответ правильный  
    System.out.println("Ответ правильный");  
}  
else { // если значение переменной answer не будет равно 13  
    // в консоль будет выведено: Ошибка в ответе  
    System.out.println("Ошибка в ответе");  
}
```

## Несколько условий else if

используются, если необходимо добавить новые варианты условий.

Каждое новое условие будет проверяться только,  
если предыдущие условия ложны

Синтаксис:

```
if (условие 1) {  
    код, который выполнится, если условие истинно  
} else if (условие 2){  
    код, который выполнится, если условие 1 ложно и условие 2 истинно  
} else {  
    код, который выполнится, если условие 1 ложно и условие 2 ложно  
}
```

## Несколько условий else if (пример)

Например,

```
if (answer == 13) { // если значение переменной answer будет равно 13
    // в консоль будет выведено: Ответ правильный
    System.out.println("Ответ правильный");
} else if (answer < 13) { // если значение переменной answer будет меньше 13
    // в консоль будет выведено: Попробуйте число больше
    System.out.println("Попробуйте число больше");
} else { //если все условия будут ложными
    // в консоль будет выведено: Попробуйте число меньше
    System.out.println("Попробуйте число меньше");
}
```

# Конструкция switch

может выполнять проверку только равенства  
(оператор if может вычислять результат булева выражения).

Две **константы** case в операторе switch не могут иметь одинаковые значения

Синтаксис,

```
switch(выражение) {  
    case значение1 : // if (выражение == значение1)  
        ... код  
        [break]  
    case значение2 : // if (выражение == значение1)  
        ... код  
        [break]  
    case значение3 : // if (выражение == значение3)  
    case значение4 : // if (выражение == значение4)  
        ... код  
        [break]  
    ...  
    default :  
        ... код, который необходимо выполнить,  
        если ни один case не совпал  
        [break]  
}
```

**Выражение** проверяется на равенство 1му значению **значение1**, затем 2му **значение2** и так далее.

Если **соответствие установлено** – switch **начинает выполняться** от соответствующего case и далее, **до ближайшего break** или **до конца switch**.

Если **ни один case не совпал** – выполняется вариант **default** (если он описан).

## Конструкция switch (пример)

```
String item = "какое-то значение";
switch (item) {
    case "Oranges": // если item == "Oranges",
        // отработает данный case и в консоле мы увидим "Oranges - $0.59 a pound."
        System.out.println("Oranges - $0.59 a pound.");
        // так как break не указан следующий case отработает без проверки условия
        // в консоле мы увидим также "Apples - $0.32 a pound."
    case "Apples": // если item == "Apples",
        // отработает данный case и в консоле мы увидим "Apples - $0.32 a pound."
        System.out.println("Apples - $0.32 a pound.");
        break;
    case "Mangoes": // если item == "Mangoes" или / и
    case "Papayas": // если item == "Papayas"
        // в консоле мы увидим "Mangoes and papayas are $2.79 a pound."
        System.out.println("Mangoes and papayas are $2.79 a pound.");
        break;
    default: // если item не найдет совпадений,
        // в консоле мы увидим "Sorry, we are out of " + item + "."
        System.out.println("Sorry, we are out of " + item + ".");
}
```