

# Тернарный оператор

Привет!

Сегодня вы узнаете о способе сокращать до одной строки программного кода простые условия, в которых есть и оператор `if`, и оператор `else`.

Представим, что необходимо сформировать сообщение в зависимости от выполнения или невыполнения какого-то условия.

Например, это сообщение над какой-нибудь пропускной системой, которое будет зависеть от двух факторов: действителен ли пропуск и не закрыто ли помещение на карантин.

Изначально будут две переменные типа `boolean`:

```
boolean passIsValid = true;
boolean inCarantine = true;
```

Если воспользоваться операторами `if` и `else`, то код можно написать такой:

```
String message;
if (passIsValid && !inCarantine) {
    message = "Проход открыт";
} else {
    message = "Проход заблокирован";
}
```

Это достаточно стандартный код, поэтому во многих языках программирования придумали простой вариант его написания. Он называется тернарным оператором. Пишется следующим образом:

```
String message = passIsValid && !inCarantine ?
    "Проход открыт" :
    "Проход заблокирован";
```

Тернарный оператор состоит из трёх частей (отсюда его название): из условия, которое может быть как переменной, так и целым выражением, и из двух результатов выполнения оператора (если условие выполняется и если нет).

Частей должно быть всегда три. Так писать нельзя:

```
String message = passIsValid && !inCarantine ?  
    "Проход открыт";
```

Разберём, почему не стоит делать несколько вложенных тернарных операторов. Например, появился еще один фактор — относится ли посетитель к обслуживающему персоналу:

```
boolean isPersonnel = true;
```

Тогда добавляется ещё одно условие:

```
String message;  
if (passIsValid && !inCarantine) {  
    message = "Проход открыт";  
} else if (isPersonnel) {  
    message = "Проход открыт только для персонала";  
} else {  
    message = "Проход заблокирован";  
}
```

В этот момент появляется соблазн написать так:

```
String message = passIsValid && !inCarantine ?  
    "Проход открыт" :  
    (isPersonnel ?  
        "Проход открыт только для персонала" :  
        "Проход заблокирован");
```

Такой код работает, но не прост для понимания. Лучше оставить цепочку операторов if и else.

```
String message;  
if (passIsValid && !inCarantine) {  
    message = "Проход открыт";  
} else if (isPersonnel) {  
    message = "Проход открыт только для персонала";  
} else {  
    message = "Проход закрыт";  
}
```

## Итоги

- Тернарный оператор — это специальный оператор, предназначенный для упрощения кода. Используется, когда нужно проверить выполнение условия и выдать результат в зависимости от того, выполнилось оно или нет.

var value = **condition** ? **yes** : **no**

Тернарный оператор состоит из трёх частей. Во-первых, это само условие, оно обозначено чёрным цветом. После условия, которое может быть булевой переменной или выражением, пишется знак вопроса. После знака вопроса через двоеточие пишутся два варианта того, что должно быть результатом выполнения тернарного оператора. На первом месте пишется значение, которое будет результатом при выполнении условия, а на втором — при невыполнении.

- Тернарный оператор, в отличие от пары операторов if-else, обязательно должен возвращать значение.
- Вложенности тернарных операторов следует избегать.