3\_4 код-лайфхаки

**1. Выражения vs Команды**

Все команды в Java делятся на две категории: команды (Statement) и выражения (Expression). Про команду обычно говорят, что она выполняется, а про выражение — что оно вычисляется. Но это не главное.

Основное отличие между командой и выражением в том, что у выражения есть результат вычисления. И у этого результата, во-первых, есть тип, а во-вторых, этот результат можно куда-то присвоить, ну или использовать в каком-то другом выражении.

Примеры:

|  |  |
| --- | --- |
| **Код** | **Примечания** |
| int x; | Команда |
| (a < 10) | Выражение, тип boolean |
| i++; | Выражение, тип совпадает с типом переменной i |
| x = 5; | Выражение, тип совпадает с типом переменной x |

И что это нам дает?

Во-первых, мы можем использовать тот факт, что очень много команд на самом деле являются выражением (имеют результат вычисления). Например, такой код будет работать:

|  |  |
| --- | --- |
| **Код** | **Примечания** |
| int x, y, z;  x = y = z = 1; | int x, y, z;  x = (y = (z = 1)) |

Во-вторых, мы можем игнорировать результат вычисления выражения, если хотим.

|  |  |
| --- | --- |
| **Код** | **Код, где мы игнорируем результат:** |
| int x = scanner.nextInt();  boolean m = (5 < 10); | scanner.nextInt();  (5 < 10); |

Игнорировать результат вычисления выражения нужно, например, если выражение кроме результата делает еще что-то полезное, и нам это действие важно, а сам результат — нет.

**2. Тернарный оператор**

Этот лайфхак уже поинтереснее предыдущего. В Java есть специальный тернарный (тройной) оператор. Чем-то он похож на сокращенную запись оператора if–else:

Условие ? Выражение1 : Выражение2;

Если условие истинно, то выполняется выражение1 иначе выполняется выражение2. После условия следует знак вопроса, а два выражения разделены двоеточием.

Основное отличие тернарного оператора от оператора if-else в том, что тернарный оператор является выражением, а значит его результат можно чему-нибудь присвоить.

Например, мы хотим вычислить минимум двух чисел. С использованием тернарного оператора этот код будет выглядеть так:

int a = 2;

int b = 3;

int min = a < b ? a : b;

Или, допустим, вам нужно присвоить переменной разные значения в зависимости от какого-то условия. Как это сделать?

Вариант первый — воспользоваться оператором if-else:

int age = 25;

int money;

if (age > 30)

money = 100;

else

money = 50;

Второй вариант — использовать тернарный оператор, то есть сокращенную запись оператора if-else:

int age = 25;

int money = age > 30 ? 100 : 50;

Так что же лучше использовать: оператор if-else или тернарный оператор? С точки зрения скорости выполнения, особой разницы нет. Здесь скорее вопрос читаемости кода. И это очень важный момент: код должен не только правильно работать, но и легко читаться другими программистами.

Самый простой критерий может быть таким: если код помещается в одну строку — используйте тернарный оператор, если в одну строку он уже не помещается — лучше использовать if-else.

**3. Сравнение вещественных чисел**

Как уже говорилось раньше, просто так сравнивать вещественные числа нельзя: в процессе работы с ними всегда могут возникать неожиданные эффекты из-за того, что часть значащих цифр отбрасывается.

Поэтому есть проверенный временем подход. Если два вещественных числа отличаются на очень маленькое значение, значит можно считать их равными. Пример:

double a = 1.000001;

double b = 1.000002;

if ( (b - a) < 0.0001 )

System.out.println("Числа равны");

else

System.out.println("Числа не равны");

Но это еще не все: разница чисел вполне может оказаться отрицательным числом. Поэтому, чтобы такой подход работал, нужно сравнивать не просто разницу чисел, а модуль разницы чисел: |a-b|

В Java есть специальный метод для вычисления модуля числа — Math.abs():

int m = Math.abs(значение);

Поэтому исправленный вариант нашего примера выше будет выглядеть вот так:

double a = 1.000001;

double b = 1.000002;

if ( Math.abs(b - a) < 0.0001 )

System.out.println("Числа равны");

else

System.out.println("Числа не равны");