****The Java Local Variable Type Inference (LVTI)**** или кратко — тип ****var**** (идентификатор ****var**** — это не ключевое слово, а зарезервированное имя типа) был добавлен в Java 10.

Являясь 100% функцией компилятора, она не влияет на байт-код, время выполнения или производительность. В основном компилятор проверяет правую часть от оператора присваивания и, исходя из нее, определяет конкретный тип переменной, а затем заменяет им ****var****.

Когда мы используем ****var**** вместо написания явных типов, то компилятор определяет их автоматически и подставляет вместо ****var****. Но с другой стороны в результате этого людям становится труднее читать и понимать код, так как использование ****var**** может усложнить его читаемость и понимание. В большинстве случаев это происходит потому, что мы склонны смотреть на тип переменной, как на первичную информацию, а на ее имя, как на вторичную. Хотя должно быть как раз наоборот.

Избегайте:

*// AVOID*

var intNumber = 20; *// this is inferred as int*

var longNumber = 20; *// this is inferred as int*

var floatNumber = 20; *// this is inferred as int*

var doubleNumber = 20; *// this is inferred as int*

Все четыре переменные будут выведены, как ****int****. Чтобы исправить это поведение, нам нужно использовать литералы Java:

*// PREFER*

var intNumber = 20; *// this is inferred as int*

var longNumber = 20L; *// this is inferred as long*

var floatNumber = 20F; *// this is inferred as float, 20.0*

var doubleNumber = 20D; *// this is inferred as double, 20.0*

* ****var****используется для локальных переменных, где, в большинстве случаев, программирование с помощью интерфейсов используется меньше, чем в случаях с параметрами методов, возвращаемыми значениями или полями
* Область действия локальных переменных должна быть небольшой, поэтому решение проблем, вызванных переключением на другую реализацию, не должно составить больших трудностей
* ****var****воспринимает код, стоящий справа, как инициализатор, используемый для определения фактического типа. Если, в какой-то момент, инициализатор будет изменен, то определяемый тип тоже может измениться, вызвав проблемы в коде, опирающемся на эту переменную.

Вы должны избегать подобных конструкций:

*// AVOID*

var productList = **new** ArrayList<>(); *// is inferred as ArrayList<Object>*

Тип будет определен, как ****ArrayList<Object>****. Это происходит потому, что информация, необходимая для корректного определения типа, не представлена. Это приводит к тому, что будет выбран ближайший тип, который может быть совместим с контекстом происходящего. В данном случае — ****Object****.

Код ниже, с использованием ****var****также не скомпилируется. Это происходит потому, что компилятор не может определить тип по правой части:

*// explicit type work as expected***int**[] numbers = {1, 2, 3};*// IT DOESN'T COMPILE*

var numbers = {1, 2, 3};

var numbers[] = {1, 2, 3};

var[] numbers = {1, 2, 3};

****Пункт 10: var нельзя использовать при объявлении нескольких переменных в одной строке****

Если вам нравится объявлять переменные одного типа разом, то вам нужно знать что ****var****не подходит для этого. Следующий код не скомпилируется:

*// IT DOESN'T COMPILE// error: 'var' is not allowed in a compound declaration*

var hello = "hello", bye = "bye", welcome = "welcome";

Вместо этого используйте:

*// PREFER*

String hello = "hello", bye = "bye", welcome = "welcome";

Или это:

*// PREFER*

var hello = "hello";

var bye = "bye";

var welcome = "welcome";

<https://topjava.ru/blog/26-items-for-dissecting-java-local-variable-type-inference>