Переменные

Привет! Сегодня вы узнаете, что такое переменные, как они создаются, какими бывают и что такое строгая (статическая) типизация.

Рассмотрим это на примере проекта Filling Station.

Программа сначала выводит в консоль фразу «Система расчёта стоимости топлива», а затем в ней создаётся сразу несколько переменных. Переменная — это место, в котором хранятся какие-то данные.

```
System.out.println("Система расчёта стоимости топлива");
int fuetType = 678;
int amount = -98;
```

Int, double

00:30–01:59

Первая такая переменная — fuelType, в ней хранится число, обозначающее тип топлива. Обратите ещё раз внимание, что перед названием переменной написано int.

Оно означает, что в этой переменной могут храниться только целые числа. Целые числа — это, например, 10, 5247, 0 или −7242789. Но числа также бывают нецелыми, с точкой. Например, 2,5, −7,89 или число пи.

Ниже как раз созданы переменные типа double:

```
double fuel92price = 60.2;
double fuel95price = 67.33;
```

Это переменные, содержащие числа с плавающей точкой. Если вы попробуете число с плавающей точкой положить в переменную int, у вас ничего не получится: если запустить такой код, то возникнет ошибка incompatible types, (несовместимые типы). Это особенность языка Java — статическая типизация.

```
/Users/sortedmap/Development/Skillbox/Java/FillingStation/src/Main.java:5:24
java: incompatible types: possible lossy conversion from double to int
```

Статическая типизация

01:59–03:45

Статическая типизация — особенность языка, при которой переменные связываются с типом в момент их объявления, и заданный тип не может быть изменён позже.

Ещё одна полезная особенность переменных в том, что вы можете как просто задавать их значения, так и писать какие-нибудь выражения справа от знака равенства.

Например, вы хотите давать скидку всем, кто купил больше 10 литров топлива. Вы можете вычесть эту скидку при присвоении значения:

```
if (amount > 10) {
fuelPrice = fuelPrice - 0.1 * fuelPrice;
}
```

Скидка будет 10%. Такие значения лучше выделять в отдельные переменные:

```
double discount = 0.1;
```

Использовать в коде эти переменные, а не числа:

```
fuelPrice = fuelPrice - discount * fuelPrice;
```

Можно написать проще:

```
fuelPrice = (1 - discount) * fuelPrice;
```

В данном случае мы переменной присваиваем не число, а целое выражение. Перед тем как итоговое число попадёт в переменную, сначала будет посчитано это выражение.

Вы познакомились с переменными типов int и double, но в Java есть и другие типы переменных. Приложения часто бывают многоязычными, и при их разработке для всех надписей в интерфейсе делают переводы на несколько языков. Для этого, все такие сообщения должны быть сначала выделены в

переменные, значения которых устанавливаются при запуске приложения или при смене языка.

String

03:45–06:51

В этом приложении есть несколько текстовых сообщений, которые можно выделить в отдельные переменные — строки. По-английски «строка» переводится как string.

Давайте выделим в такую переменную сообщение о том, что указан неверный тип топлива:

```
String wrongFuelTypeMessage = "Указан неверный тип топлива";
```

Используем эту переменную в коде:

```
        System.out.println(wrongFuelTypeMessage);
```

В строку можно выделить и более сложное значение:

```
String fuelPriceMessage = "Цена выбранного топлива: " + fuelPrice
+ " руб.";
System.out.println(fuelPriceMessage);
```

При формировании этого сообщения используется переменная fuelPrice. Поскольку в Java программный код выполняется построчно сверху вниз, то использовать переменные можно только после их создания.

К примеру, строку fuelPriceMessage можно задать только после того, как создана переменная fuelPrice. Если этот код перенести до строки int fuelPrice, то работать он не будет.

```
String wrongFuelTypeMessage = "Указан неверный тип топлива";
String fuelPriceMessage = "Цена выбранного топлива: " + fuelPrice + " руб.";
```

При этом мы можем разделить объявление переменной и присвоение ей значения. Сначала написать:

```
      String fuelPriceMessage;
```

А когда переменная fuelPrice уже будет объявлена, задать значение строки:

```
      fuelPriceMessage = "Цена выбранного топлива: " + fuelPrice +
" руб.";
```

Таким же образом мы можем задать переменные, содержащие и все остальные сообщения, используемые в коде.

При создании строки её текст задаётся в двойных кавычках. По кавычкам компилятор понимает, что написано: код, который нужно выполнять, или просто текст.

Boolean

06:51–09:05

Еще один тип переменной — boolean. Переменные этого типа могут принимать всего два значения — true или false («истина» или «ложь»).

Переменные такого типа обычно используются для обозначения чего-то, что может быть в двух состояниях. Например, вы хотите, чтобы система расчёта стоимости заправки учитывала наличие скидки у конкретного покупателя. В этом случае целесообразно создать переменную типа boolean. Назовём её hasDiscount («есть скидка»):

```
boolean hasDiscount = true;
```

Значение true («истина») будет говорить о том, что скидка есть. Если значение будет равно false («ложь»), то это будет говорить о том, что скидки нет:

```
boolean hasDiscount = false;
```

Используем эту переменную. Ранее мы давали скидку всем, кто купил более 10 литров топлива. А теперь будем давать всем, у кого просто есть эта скидка:

```
if (hasDiscount) {
fuelPrice = fuelPrice - 0.1 * fuelPrice;
}
```

В условии if пишется выражение, которое в результате выполнения превращается в переменную типа boolean — то есть оно в итоге равно либо true, либо false.

Можно вернуть код обратно, но выделить проверяемое условие в переменную:

```
        boolean hasDiscount = amount > 10;
```

Переменная hasDiscount будет равна true только в том случае, если выражение «amount > 10» будет выполняться, то есть значение переменной amount будет больше десяти.

С более сложными выражениями, результатом которых является переменная типа boolean, а также с более сложными вариантами использования условных операторов (в частности, оператора if), мы ещё познакомимся в следующих темах этого модуля.

Ключевое слово var

09:05–11:14

В Java есть более простой способ задавать переменные, который появился в 10-й версии языка.

Это слово var. Используя это слово, вы можете задать переменную таким образом:

```
var hasDiscount = amount > 10;
```

Так же можно задавать и переменные любых других типов:

```
var wrongFuelTypeMessage = "Указан неверный тип топлива";
var discount = 0.1;
var fuelType = 92;
```

Тип переменных при этом будет определяться автоматически. Но поскольку в Java статическая типизация, поменять его по-прежнему будет нельзя:

```
fuelType = "тип топлива";
```

Будет возникать всё та же ошибка несоответствия типов. Более того, нельзя с помощью слова var сначала объявить переменную, а потом задать её значение:

```
var fuelPriceMessage;
```

Такой код работать не будет.

Java отличается от ряда других языков программирования не только статической, но и явной типизацией. Явная типизация — это свойство языка программирования, при котором тип переменной объявляется в явном виде, то есть перед тем как написать название переменной, мы указываем, какого она будет типа — int, double, string, boolean или какого-то ещё. И слово var — это «синтаксический сахар» — некое упрощение, позволяющее быстрее перейти на Java разработчикам, которые писали код на языках с неявной типизацией, например Python или JavaScript.

Java — компилируемый язык программирования, и весь код, который пишут разработчики, перед исполнением Java-машиной превращается в так называемый байт-код. В байт-коде типы переменных всегда определены в явном виде, независимо от того, объявляли вы их явно или через слово var. Но это нововведение пока не устоялось. Разработчики на Java, как правило, его не используют.

Итоги

с 11:14 до конца

Вы познакомились с некоторыми типами переменных в Java:

- int — целые числа,
- double — числа с плавающей точкой,
- boolean — true или false,
- String — строка.

В Java больше типов переменных. Полную классификацию мы будем изучать детально в следующих модулях курса. А пока вам будет достаточно этих типов.

Также вы познакомились с тем, как можно задавать переменные: простым присвоением значения или написанием какого-либо выражения; можно сначала объявить переменную, а потом отдельно задать её значение.

Для объявления переменных также можно использовать слово var, но это не принято в среде Java-разработчиков и никак не влияет на используемую в Java типизацию, статическую и явную. С этими понятиями мы также познакомились.

Вы узнали о том, что выражения, проверяемые при помощи условного оператора if, в результате выполнения становятся равны либо true, либо false и их можно выделять в переменные типа boolean.