

Переменные и константы

Переменные используются для хранения используемых в программе данных. Любая переменная имеет три базовых характеристики:

- имя – уникально идентифицирует переменную и позволяет к ней обращаться в программе;
- тип – определяет внутреннее представление данных в памяти компьютера, множество значений, которые могут принимать величины этого типа, и операции, которые могут выполняться над величинами этого типа;
- значение – текущая величина, хранящаяся в переменной на данный момент.

Работа с переменной всегда начинается с ее *объявления*, которое включает в себя имя и тип объявляемой переменной. Значение переменной также может быть указано при её объявлении (это называется *инициализацией*), но в большинстве случаев задание начального значения можно отложить. В роли инициализатора переменной может выступать не только литерал, но и арифметическое или логическое выражение, результат которого одинаков или совместим с типом переменной. Если это выражение может быть вычислено во время компиляции, то компилятор сразу записывает в переменную результат вычисления, в противном случае это действие откладывается на момент выполнения программы.

Таким образом, при описании переменной сначала указывается тип, затем ее имя и, если необходимо, инициализатор, который может быть константой или выражением, вычисляемым во время компиляции или исполнении программы. В частности, можно использовать и объявленные ранее переменные. Далее можно поставить запятую и объявить новую переменную точно такого же типа.

После объявления переменная может быть использована в различных выражениях, в которых будет браться ее текущее значение. Также в любой момент можно изменить значение переменной, используя оператор присваивания так же, как это делалось в инициализаторах.

Кроме того, при объявлении переменной может быть использовано ключевое слово `final`. Его указывают перед типом переменной, которую в этом случае необходимо сразу проинициализировать и впоследствии никогда не менять ее значение. Если попытаться изменить значение `final`-переменной, это приведет к ошибке компиляции. Таким образом, `final`-переменные становятся чем-то вроде констант, но, на самом деле, некоторые инициализаторы могут вычисляться только во время исполнения программы.

Область видимости

Java является объектно-ориентированным языком программирования, поэтому любую программу, написанную на Java, можно представить как набор взаимодействующих между собой *объектов*. *Класс* является шаблоном или описанием такого объекта, а сам объект представляет собой *экземпляр класса*.

Любой объект может обладать двумя основными характеристиками: *состояние* - некоторые данные, которые хранит объект, и *поведение* - действия, которые может совершать объект. Для хранения состояния объекта в классе применяются *поля*, или переменные класса. Для определения поведения объекта в классе применяются *методы*.

Чтобы не заставлять программистов, совместно работающих над различными классами одной системы, координировать имена, которые они дают различным конструкциям языка, у каждого имени есть область видимости. *Областью видимости переменной* является фрагмент программы от точки ее описания до конца текущего блока. *Блоком* в языке Java называют фрагмент программы, который начинается открывающей фигурной скобкой и заканчивается закрывающей фигурной скобкой. *Время жизни переменной* в Java определяется правилом: переменная создается в точке ее описания и существует до момента окончания того блока, в котором находится данное описание. Время жизни переменных в языке Java совпадает с их областью видимости с учетом отличия самих этих понятий.

В различных языках программирования существуют различные типы или классы переменных – локальные, глобальные, статические и т. д. В языке Java существует только один тип переменных – *локальные переменные*, действующие в пределах одного метода. Объявление локальных переменных возможно в любом месте метода, при этом областью видимости переменной будет тот блок, в котором она определена. Если в блок, где описана данная переменная, вложены другие блоки, то переменная будет доступна и в них. Но, в отличие от многих других языков программирования, в языке Java запрещено *переопределять* переменную во вложенных блоках, т.е. описывать другую переменную с тем же именем. Следовательно, имя переменной в пределах её области видимости должно быть уникальным.

Совсем иначе дело обстоит с объектами. Объекты доступны в программе только через ссылки на них, поэтому *область видимости объекта* определяется областью видимости ссылок на этот объект (на один объект может быть сколько угодно ссылок). *Время жизни объекта* определяется следующим правилом: объект существует, пока существует хотя бы одна ссылка на него. Как только пропадет последняя ссылка на объект, он становится недоступным и может быть уничтожен сборщиком мусора.