**2.5 Java Naming conventions** :-

Java **naming convention** is a rule to follow as you decide what to name your identifiers such as class, package, variable, constant, method etc.

But, it is not forced to follow. So, it is known as convention not rule.

All the classes, interfaces, packages, methods and fields of java programming language are given according to java naming convention.

Advantage of naming conventions in java

By using standard Java naming conventions, you make your code easier to read for yourself and for other programmers. Readability of Java program is very important. It indicates that **less time**is spent to figure out what the code does.

|  |  |
| --- | --- |
| **Name** | **Convention** |
| class name | should start with uppercase letter and be a noun e.g. String, Color, Button, System, Thread etc. |
| interface name | should start with uppercase letter and be an adjective e.g. Runnable, Remote, ActionListener etc. |
| method name | should start with lowercase letter and be a verb e.g. actionPerformed(), main(), print(), println() etc. |
| variable name | should start with lowercase letter e.g. firstName, orderNumber etc. |
| package name | should be in lowercase letter e.g. java, lang, sql, util etc. |
| constants name | should be in uppercase letter. e.g. RED, YELLOW, MAX\_PRIORITY etc. |

## CamelCase in java naming conventions

Java follows camelcase syntax for naming the class, interface, method and variable.

If name is combined with two words, second word will start with uppercase letter always e.g. actionPerformed(), firstName, ActionEvent, ActionListener etc.

Java Naming Conventions

Below are some naming conventions of for java programming language. They must be followed while developing software in java for good maintenance and readability of code. Java uses CamelCase as a practice for writing names of methods, variables, classes, packages and constants.

**Camel case in Java Programming :** It consists of compound words or phrases such that each word or abbreviation begins with a capital letter or first word with a lowercase letter, rest all with capital.

1. **Classes and Interfaces** :
   * Class names should be **nouns**, in mixed case with the **first** letter of each internal word capitalized. Interfaces name should also be capitalized just like class names.
   * Use whole words and must avoid acronyms and abbreviations.

Examples:

Interface Bicycle

Class MountainBike implements Bicyle

Interface Sport

Class Football implements Sport

1. **Methods :**
   * Methods should be **verbs**, in mixed case with the **first letter lowercase** and with the first letter of each internal word capitalized.

Examples:

void changeGear(int newValue);

void speedUp(int increment);

void applyBrakes(int decrement);

1. **Variables :**Variable names should be short yet meaningful.
   * Should **not** start with underscore(‘\_’) or dollar sign ‘$’ characters.
   * Should be mnemonic i.e, designed to indicate to the casual observer the intent of its use.
   * **One-character variable names should be avoided** except for temporary variables.
   * Common names for temporary variables are i, j, k, m, and n for integers; c, d, and e for characters.

Examples:

// variables for MountainBike class

int speed = 0;

int gear = 1;

1. **Constant variables:**
   * Should be **all uppercase** with words separated by underscores (“\_”).
   * There are various constants used in predefined classes like Float, Long, String etc.

Examples:

static final int MIN\_WIDTH = 4;

// Some Constant variables used in predefined Float class

public static final float POSITIVE\_INFINITY = 1.0f / 0.0f;

public static final float NEGATIVE\_INFINITY = -1.0f / 0.0f;

public static final float NaN = 0.0f / 0.0f;

1. **Packages:**
   * The prefix of a unique package name is always written in **all-lowercase ASCII letters** and should be one of the top-level domain names, like com, edu, gov, mil, net, org.
   * Subsequent components of the package name vary according to an organization’s own internal naming conventions.

Examples:

com.sun.eng

com.apple.quicktime.v2

// java.lang packet in JDK

java.lang

This article is contributed by **Gaurav Miglani**. If you like GeeksforGeeks and would like to contribute, you can also write an article using[contribute.geeksforgeeks.org](http://www.contribute.geeksforgeeks.org/) or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.