**Variable, constant and methods**

**Identifier**

Identifiers are used to define names in programmes. For example, class names, variable names and method names are all called identifiers.

**Regarding the definition of Java identifiers, please note:**

* All identifiers should be started with letters (A-Z/a-z), the symbol of US dollar ($) or underline(\_).
* Characters should be followed with any character combinations of letters (A-Z/a-z), the symbol of US dollar ($), underline(\_) and numbers.
* Keywords cannot be used as identifiers.
* Identifiers are case sensitive.
* Examples of legitimate identifiers*: age, $salary, \_value, \_\_1\_value*Examples of illegitimate identifiers\*：123abc、-salary

**Keywords**

We just mentioned above that keywords cannot be used as identifiers. Below are some of the keywords in Java language:

| **int** | **short** | **try** | **char** | **final** | **interface** | **static** | **void** | **float** | **native** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| abstract | continue | for | new | switch | assert | default | goto | super | while |
| package | synchronized | boolean | do | if | private | this | break | class | finally |
| double | implements | protected | throw | byte | else | import | public | long | stricfp |
| throws | case | enum | instanceof | return | transient | catch | extends | volatile | const |

*Note:*

* "const" and "go" are no longer used as keywords, but they are still listed here.
* Many people wrongly believe that true，false and null are keywords, but actually they are not. They are just literals.

**Variables**

A variable is a value able or apt to vary. In the context of Java, there are several kinds of variables:

* Local variables: there are no special keywords to mark local variables. Its local nature is determined by its location, which should be between the brackets of methods. Therefore, a local varialbe is only visible to the method declaring it, and invisible to other parts of a class.
* Class variables (static field): class variables are fields modified by the key word of static. It will tell a compiler no matter how many instances a class has, the variable only has one copy
* Member variables (non-static field): Techinically speaking, an object will save its status in the non-static field, namely the field which is not modified by the keyword of static. Non-static field is also called instance variable. That is because each instance of a class(in other words, each object) is independdent.

**Naming conventions of variables:**

* Variables are identifiers, so its naming should follow the conventionss of identifiers.
* A name of a variable should be meaningful. Its meaning should be expressed explicitly.
* It's recommended to use English nouns as variable names. Pinyin is not allowed.
* Lower Camel Case should be applied, where the first letter of the entire word is lowercase, but subsequent first letters are uppercase. It is a way to distinguish adjacent words, such as name, bookTitle etc.

For related naminng conventions, please visit [here](https://www.oracle.com/technetwork/java/codeconventions-135099.html)

**Constants**

Constants are usually used to refer to values unable to change. In softwre development you may find some values unable to be changed once defined, such as pi etc. For such cases, constants should be used and declared using the keyword of final.

**Regarding the understanding of constants, the below points should be noted:**

* When defining a constant, you should first initialize it.
* Once you have finished initializtion, you cannot assign another value to a constant.
* The keyword of final can not only modify constants of basic data, but also modify the refernce and method of an object.
* The keyword of final should be used simultaneously with static.

**Naming conventions of constants**

* Constants are identifiers, so its naming should follow the conventions of identifiers.
* A name of a constant should be meaningful. Its meaning should be expressed explicitly.
* It's recommended to use English nouns as constant names. Pinyin is not allowed.
* Each letter of constant names should be capitalized, with two adjacent words divided by an underline (\_). For example, final int PI = 3.1415927; etc.

For related naming conventions, please refer to [here](https://www.oracle.com/technetwork/java/codeconventions-135099.html)

**Methods (Function)**

Methods are sets of codes which have independent business logic. It is usually an orderly combination of steps to solve a certain problem. Methods are included in objects, creaated in programme and called in other places.

**Definition**

modifier return type name of **methods** (parameter type parameter name) {

method type

**return** **return** value;

}

* Modifier: it can be selected to tell compiler how to call a certain method. It is used to define the access type of this method.
* Return type: it refers to the type of return value. For those which don't have return values, you can use the keyword of void to indicate.
* Name of methods: it refers to the name of methods, which should follow certain naming conventions.
* Parameter: it is composed of parameter type and parameter names. As a variable, parameter is used to receive the data transderred during calling methods.
* Method body: it refers to independent code block which has been encapsulated. It is the core of methods and is responsible for processing important business logic.

For example:

**public** **int** **getMaxValue**(**int** val1, **int** val2) {

**int** result = val1 >= val2 ? val1 : val2;

**return** result;

}

**Calling methods**

It is quite simple to call a method. You just need to call the method name and transfer the corresponding parameter. For example, the method calling for above should be:

**int** maxValue = getMaxValue(2, 3);

**Naming conventions of methods:**

* As identifiers, method names should follow the naming conventions of identifiers.
* The names of methods should have explicit meanings.
* It is better to use English verbs as method names to indicate an action. Chinese pinyin is not allowed.
* Method names should apply Lower Camel Case where the initial letter is in lower case and the first letter of the adjacent word is in upper case to differentiate.

For related naming conventions please refer to [here](https://www.oracle.com/technetwork/java/codeconventions-135099.html)

*Highly recommended: method body conforms to the principle of single responsibility where each method only does one thing. The aim is to improve readability and maintainability.*