

Forum:[Hack Forums](#)

Thread: [Basic Java Syntax and Tutorial - By Insight | Written by Bald Cat](#)

Balding - September 5, 2015, 4:40 pm

Basic java syntax - written by Bald Cat hosted by Insight

Table of Contents:

Basic Syntax
Java Identifiers
Java Modifiers
Java Arrays
Credits

Objects in Java

An object is defined as follows: An object is an entity that has state or behavior.

The characteristics broken down:

- State
- Behavior
- Identity

1. State: The state represents data or a value of an object.
2. Behavior: Behavior represents the function of an object. Ex: Pull, withdraw, deposit, etc.
3. Identity: Object identity is typically implemented via a unique ID. The value of the ID is not visible to the user. Instead, it is used by the JVM to identify each object.

Classes in Java

A class is a what objects are created from. A class can contain the following:

- Method
- Constructor
- Block
- Interface

What do they do?

1. Method: The method is a function. It gives action to an object.
2. Constructor: The constructor initializes the object.
3. Block: The block is used to initialize the instance data.
4. Interface: The interface is essentially a blueprint of the class.

```
Code public class Classexample {  
    public int this;  
    public int is;  
    public int a;  
    public int class;  
}
```

The code above obviously won't work for anything I'm just using it as an example.

Methods in Java

A method is a few lines of code which is used by name and can be called at any point in a program simply by calling the name. Methods are usually used to save time when writing code. They're very efficient.

```
Code public static int methodName(int a, int b) {  
    // body  
}
```

Instance Variables in Java

An instance variable is a variable defined in a class. Each object in a class has a separate instance.

```
Code class BaldCat{  
  
    public String realName;  
    public String[] streetAddress;  
    public int age;  
}
```

Java Identifiers

Names used for classes, variables and methods are called identifiers. Below are some important identifier rules that you should always remember.

- All identifiers should begin with a letter (A to Z or a to z). You can also begin them with (\$) or an underscore (_)
- A key word cannot be used as an identifier.
- This one is most important - identifiers are CASE SENSITIVE!! Always remember this.

Java Modifiers

Just like other languages, Java also has modifiers. There are two types of modifiers in Java.

- Access Modifiers
- Non access Modifiers

1. Access modifiers: Used to set access levels for classes, methods, constructors, and variables.

The four access levels are shown as follows:

- Visible to the package. This is default. No modifiers are needed. Basic visibility.
- Visible to the class only (private).
- Visible to the world (public).
- Visible to the package and all subclasses (protected).

Private access example

```
Code public class Logger {  
    private String format;  
    public String getFormat() {  
        return this.format;  
    }  
    public void setFormat(String format) {  
        this.format = format;  
    }  
}
```

Public access example

```
Code public static void main(String[] arguments) {  
    // rest of code goes here  
}
```

Private access example

```
Code class AudioPlayer {  
    protected boolean openSpeaker(Speaker sp) {  
        // implementation details  
    }  
}
```

```
class StreamingAudioPlayer {  
    boolean openSpeaker(Speaker sp) {  
        // implementation details  
    }  
}
```

2. Non access modifiers: Java provides a number of non-access modifiers to achieve many other functionality. Below are a few non access modifiers.

- The static modifier for creating class methods and variables.
- The final modifier for finalizing the implementations of classes, methods, and variables.
- The abstract modifier for creating abstract classes and methods.
- The synchronized and volatile modifiers, which are used for threads.

Java Arrays

An array stores a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.

Declaring array variables

Code dataType[] arrayRefVar;

Creating new arrays

Code arrayRefVar = new dataType[arraySize]; The code above does two things:

1. It creates an array using new dataType[arraySize];

2. It assigns the reference of the newly created array to the variable arrayRefVar.

Declaring an array variable, creating an array, and assigning the reference of the array to the variable can be combined in one statement, as shown below:

Code `dataType[] arrayRefVar = new dataType[arraySize];` Or

Code `dataType[] arrayRefVar = {value0, value1, ..., valuek};`

Credits

Writer/Compiler: Bald Cat

Group leader: Hinch

Misc help: <http://www.tutorialspoint.com/java/>

Hinch - September 5, 2015, 4:43 pm

Nice high quality tutorial from a fellow insight member