

Encapsulation vs Abstraction in Java

📅 Last Updated: January 29, 2022 👤 By: Lokesh Gupta 📁 Java Object Oriented Programming 🔖 Abstraction, Encapsulation, Java OOP

Most of you will agree that encapsulation and abstraction together brings a lot of confusion. Most blogs add only confusion further. Lets solve this puzzle.

I started working on this post after my previous post "[Understanding abstraction in java](#)". My goal was to understand encapsulation in java and how it relates to abstraction. As soon as i began, i started going down in more confusion like never before. After browsing many hours and reading some really well written as well as confusing blog entries, i was able to make out some clear understanding. Follow my footprints..

Table of Contents

[Encapsulation in simple words](#)

[Encapsulation in Detail](#)

[Encapsulation vs Abstraction](#)

Encapsulation in simple words

Wrapping data and methods within classes in combination with implementation hiding (through access control) is often called encapsulation. The result is a data type with characteristics and behaviors. **Encapsulation essentially has both i.e. information hiding and implementation hiding.**

Encapsulation in Detail

I read it somewhere : "**Whatever changes, encapsulate it**". It has been quoted as a famous design principle. For that matter in any class, changes can happen in data in runtime and changes in implementation can happen in future releases. So, encapsulation applies to both i.e. data as well as implementation.

Access control or implementation hiding puts boundaries within a data type or class for two important reasons. The first is to establish what the client programmers can and can't use. This feeds directly into the second reason, which is to separate the interface from the implementation.

If you are sure that client programmers can't do anything but send messages to the public interface, then you are free to change anything that's not public (e.g., package access, protected, or private) without breaking client code. Encapsulation helps you in achieving this surety.

Encapsulation vs Abstraction

If you have gone through [my last post](#), you will see that abstraction is essentially an idea, which helps in setting the guidelines. Encapsulation is the mechanism by which we achieve the desired abstraction.

In short, from OOAD perspective:

- *Abstraction is more about '**What**' a class can do. **[Idea]***
- *Encapsulation is more about '**How**' to achieve that functionality. **[Implementation]***

I have seen many contradictions to this theory over many blogs. So, if you also don't agree with this, please bear with me. Also, i will request you to put a comment you idea related to topic. I will happily try to relate or negate.

Going forward, i will take example of our well known class [HashMap](#). This class is responsible for storing key-value pair, searching based on key and do more things. From outside, client code only knows the method names and their behavior. It calls these methods and live happily. This is actually what abstraction guidelines are. Abstraction says that client code should call a method to add key-value pair, a method to retrieve value based on key and so on. How it should be done? is not business of abstraction.

And here comes encapsulation, when you start writing actual code. You write `HashMap.Entry` class and create variable *table* of type `Entry []`. Then you declare all such things private and give public access to only `put()` and `get()` methods etc. This is actually encapsulation. *A realization of your desired abstraction.*

I hope that you have more clarity about **java encapsulation** and it's **difference with abstraction**.

Happy learning !!

Was this post helpful?

Let us know if you liked the post. That's the only way we can improve.

Yes

No

Recommended Reading:

1. [Guide to Abstraction](#)
2. [Interface vs Abstract Class in Java](#)
3. [Overloading vs Overriding in Java](#)
4. [Java Access Modifiers](#)
5. [Constructors in Java](#)
6. [Java Instance Initializer Blocks](#)

- 7. [Guide to Polymorphism](#)
- 8. [Object Oriented Programming](#)
- 9. [Association, Aggregation and Composition](#)
- 0. [Guide to Inheritance](#)

Join 7000+ Awesome Developers

Get the latest updates from industry, awesome resources, blog updates and much more.

Email Address

Subscribe

** We do not spam !!*



23 thoughts on “Encapsulation vs Abstraction in Java”

Anubhav Agarwal

July 22, 2018 at 4:30 pm

nice but can you give some code on it .It will help me a lot

[Reply](#)**Viji**[May 16, 2016 at 3:08 am](#)

Can you please explain why we use interface in design patterns instead of concrete classes?

[Reply](#)**Palash Kanti Kundu**[November 17, 2015 at 7:32 am](#)

Nice article. Can you please add some code with your article ?

That would be a great help for programmers to get into the concept.

[Reply](#)**Sandip**[April 24, 2015 at 2:28 am](#)

Now i got the somewhat clear idea about encapsulation and abstraction.

[Reply](#)**Ranganath**[November 28, 2014 at 4:43 am](#)

Good explanation on abstraction and encapsulation. Thank you

[Reply](#)

Abhishek Gaur

November 17, 2014 at 1:46 pm

Nice explanation. Kindly add practical examples and code also.

I also found good article here:

<https://stackoverflow.com/questions/11966763/java-encapsulation>

[Reply](#)

swekha

September 3, 2014 at 12:04 pm

hiiii,I have one question.Let consider mobile ph. example for understanding abstraction and encapsulation.we have keypad.we don't know the internal implementation of various circuits with keypad.but we have the option to use this for dialling.this is known as encapsulation.then my question is what is abstraction here????? please give me example on mobile ph.

[Reply](#)

Lokesh Gupta

September 3, 2014 at 12:42 pm

You type a number and press/touch "DIAL" button. This dial button is actually an example of abstraction of making call.

[Reply](#)

amit

July 26, 2014 at 4:03 pm

is encapsulation and data hiding the same ?

[Reply](#)

Lokesh Gupta

July 26, 2014 at 6:35 pm

In layman term, yes data hiding is encapsulation. But, encapsulation is not only data hiding but much more than that.

[Reply](#)**Lokesh Gupta**

July 26, 2014 at 6:44 pm

I think it's good guide: <https://github.com/javaee/jaxb-v2>

[Reply](#)**HIMANSU NAYAK**

May 4, 2014 at 1:47 am

Hi Lokesh,
Well explained...
Even Single Responsibility Principal can be imposed better on class using Encapsulation

[Reply](#)**sreenath ravva**

March 24, 2014 at 5:42 am

Can we say abstraction is a thought process and encapsulation is an implementation of abstraction ???

[Reply](#)**Lokesh Gupta**

March 24, 2014 at 7:20 am

Not really. Both are parts of implementation. In fact, abstraction is mostly represented via interfaces and encapsulation via implementing classes.

[Reply](#)

sreenath ravva

[March 24, 2014 at 8:56 am](#)

ok, its clear now, Thank you!!

[Reply](#)

Shivangi Nigam

[January 24, 2014 at 10:23 am](#)

You made it easy to remember 😊

[Reply](#)

Satish

[January 2, 2014 at 4:27 pm](#)

Hi, can u please explain it with an example, not with real time objects, which we use in our programs,

Thanks in advance.

[Reply](#)

ajay

[October 29, 2013 at 8:52 am](#)

Nice explained. ...

[Reply](#)

kamini

October 4, 2013 at 5:06 pm

Can you please tell me why we need all 4 oops concepts in programing i know little but still confused.. please

[Reply](#)**Lokesh Gupta**

October 4, 2013 at 9:59 pm

You don't need to have. You write any program, and it will have all fours. These are the building blocks of language. You can't write your program like I will use only 3, and not last fourth. It does not happen.

[Reply](#)**subbareddy**

July 20, 2013 at 10:44 am

good explanation.....

[Reply](#)**subbareddy**

July 20, 2013 at 10:43 am

good explanation.....

[Reply](#)**Marko**

July 15, 2013 at 11:50 am

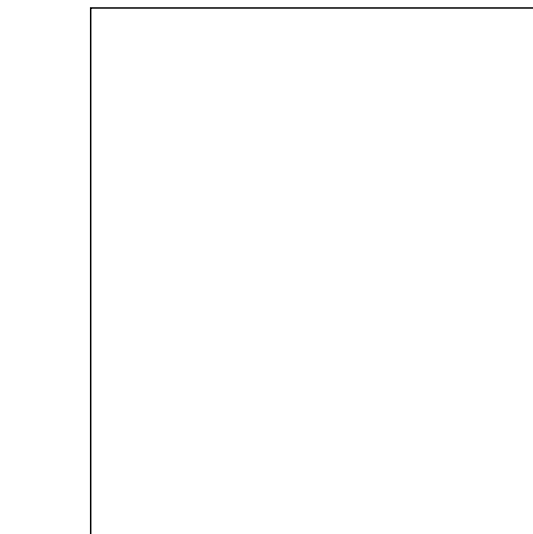
It has been a very mis-understood concept ever in java. You have concluded it very effectively and in simple words which are easy to understood by all. Great post.

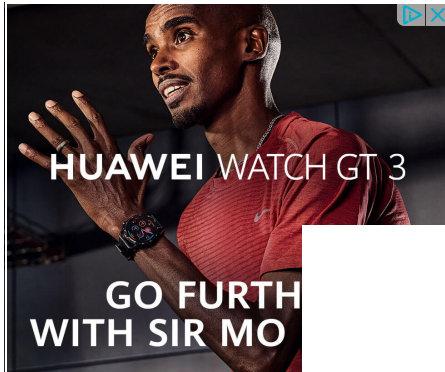
[Reply](#)

Leave a Comment

☐ Add me to your newsletter and keep me updated whenever you publish new blog posts

Post Comment





HowToDoInJava

A blog about Java and related technologies, the best practices, algorithms, and interview questions.

Meta Links

- › [About Me](#)
- › [Contact Us](#)
- › [Privacy policy](#)
- › [Advertise](#)
- › [Guest Posts](#)

Blogs

[REST API Tutorial](#)



Copyright © 2022 · Hosted on [Cloudways](#) · [Sitemap](#)