



THE MAGIC OF METAPROGRAMMING

About Me

NAME: Vergie Ericson T

EMAIL: vergieericson@gmail.com

PHONE: +62 838 4854 4451



LINKEDIN:

<https://www.linkedin.com/in/vergieet/>

GITHUB:

<https://www.github.com/vergieet>

EDUCATION



2014 - 2017 Telkom School Malang

2019 - now Mercu Buana University



EXPERIENCE

Aero Systems Indonesia

Beon Intermedia

Traveloka

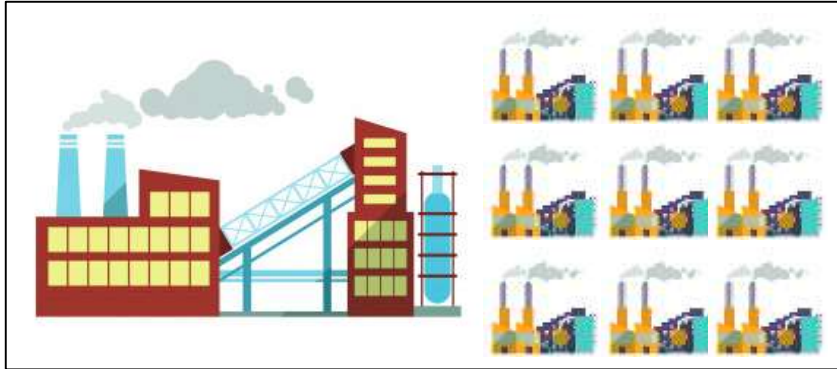
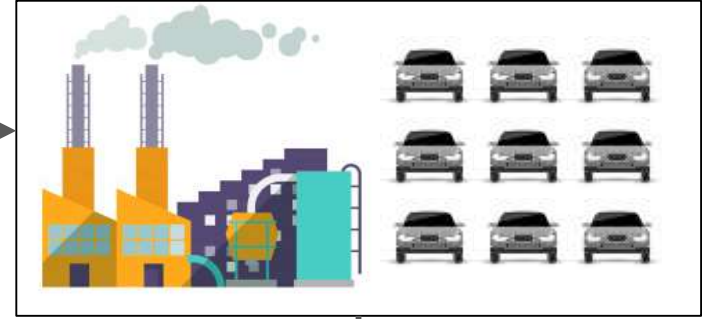
METAPROGRAMMING

Metaprogramming ways a program has knowledge of itself or can manipulate itself.

Which means programs that manipulate programs.

Imagine a guy who builds cars.

Then he builds he builds factories to build cars, and it's much better. He's now programming.



Now he decides to build factories that build factories that build cars.
That's metaprogramming.

METADATA

Metadata is "data that provides information about other data"

Which means data that describe other data.

REFLECTION

Reflection is a programming technique that allows us to examine and even modify the structure of objects at run-time.

- `getMethods()` => get all method even from inherited method
- `getDeclaredMethods()` => get declared method inside class
- `getReturnType()` => get return type of method
- etc..

```
public static void main(String []args){  
    String[] listOfString = {"1","2","3"};  
    Integer[] listOfInteger = {1,2,3};  
  
    printStrings(listOfString);  
    printNumbers(listOfInteger);  
  
    printReflection(listOfString);  
    printReflection(listOfInteger);  
}
```

```
$java -Xmx128M -Xms16M HelloWorld
```

```
String: 1 2 3
```

```
Number: 1 2 3
```

```
String: 1 2 3
```

```
Number: 1 2 3
```



```
public static void printStrings(String[] strings){  
    System.out.print("String: ");  
    for(String string : strings){  
        System.out.print(string + " ");  
    }  
    System.out.println();  
};
```

```
public static void printNumbers(Integer[] integers){  
    System.out.print("Number: ");  
    for(Integer integer : integers){  
        System.out.print(integer+ " ");  
    }  
    System.out.println();  
};
```

```
public static <T> void printReflection(T[] objects){  
    if(Integer.class.isAssignableFrom(objects.getClass().getComponentType())){  
        System.out.print("Number: ");  
    }else{  
        System.out.print("String: ");  
    }  
  
    for(T object : objects){  
        System.out.print(object+ " ");  
    }  
    System.out.println();  
};
```

ANNOTATION

Annotation is sort of tag mechanism you can label some meta on Classes, Methods or Parameters and Annotation Processing will analyze those Annotations.

Sample that already exist !

- Butterknife
- Retrofit

- Spring Boot
- Hibernate
- Ruby on Rails

- and many more ...

THANKS!

The background features a dark blue gradient with two large, wavy, red wireframe mesh structures on the left and right sides, creating a sense of depth and movement.

Github link: <https://github.com/vergieet/sample-android-metaprogramming>

SOURCE & REFERENCES

- <https://blogs.oracle.com/java/metaprogramming:-manipulating-data-about-data#:~:text=Javadoc%20is%20also%20an%20example,in%20Java%20is%20using%20annotations.&text=Metaprogramming%20is%20writing%20programs%20that,or%20themselves%20based%20on%20metadata.>
- <https://en.wikipedia.org/wiki/Metadata>
- https://medium.com/@hinchman_amanda/oop-metaprogramming-series-java-reflection-and-aop-6f63b23f0b05
- <https://stackoverflow.com/questions/2565572/metaprogramming-self-explanatory-code-tutorials-articles-books>
- Business vector created by macrovector_official - www.freepik.com
- Car vector created by macrovector - www.freepik.com
- https://www.siliconindia.com/events/siliconindia_events/presentation/FN9h1680.pdf