

-Java

1. [redacted] URLHTML

2. [redacted]

3. [redacted]

4. [redacted]

5. [redacted]

6. [redacted] UpperCamelCase :()DO / BO / DTO / VO

```
UserDAO,ParameterNameSpace,NetworkHelper
```

```
userDAO,ParameternameSpace,NETworkHelper
```

7. [redacted] lowerCamelCase

```
localValue,getPureMessage(),userTextField
```

8. [redacted]

```
public static final int MAX_STOCK_COUNT = 20;
```

```
public static final int max_stock_count = 20;  
public static final int MAX_COUNT = 20;
```

9. [redacted] Abstract; Exception ; Test

10. [redacted]

```
package com.fr.data.model;
```

```
package com.fr.chart.ChartAttr; //
package com.finebi.common.internalimp //
```

11. [REDACTED]

```
public class DateWork {
    private int remainDays = 10;//

    public void calculateNameWithData(Data data) {

    }
}
```

```
public class DateWork {
    private int r = 10;//

    public void cal(Data data) {

    }
}
```

12. [REDACTED] setis/hasget

```
public class Student {
    private boolean male;

    public boolean isMale() {
        return male;
    }
}
```

```
public class Student {
    private boolean male;

    public boolean getMale() {
        return male;
    }
}
```

13. [REDACTED] is/has

```
public class Student {
    private boolean male;

    public boolean isMale() {
        return male;
    }
}
```

```
public class Student {
    private boolean isMale;

    public boolean getMale() {
        return isMale;
    }
}
```

14. static final

```
private static final int PI = 3.14;
public double area(double r) {
    return PI * r * r;
}
```

```
public double area(double r) {
    return 3.14 * r * r;
}
```

15. Custom

```
public interface PasswordHandleProcessor extends XMLable, Immutable {
    String encrypt(String clientPassword);
    String decrypt(String clientPassword);
}
```

```
public interface CustomPasswordCryptProcessor extends XMLable, Immutable {
    String encodePassword(String clientPassword);
    String decodePassword(String clientPassword);
}
```

16. setset

```
public setIp(String ip) {  
    this.ip = ip;  
}
```

```
public BlockIpEntity ip(String ip) {  
    this.setIp(ip);  
    return this;  
}
```

```
public BlockIpEntity setIp(String ip) {  
    this.ip = ip;  
    return this;  
}
```