-Java

1. [redacted] init

```
public class Fish {
   public Fish() {
      init();
   }
   public void init() {
     // do something for init
   };
}
```

```
public class Fish {
   public Fish() {
      // do something for init
   }
}
```

2. [redacted] nullnull

```
public class Utils {
   public static Border createBorder(Font font, Color color) {
         if (color == null) {
             color = new Color(223, 122,123);
          }
    }
}
```

```
//Utils
public static Border createBorder(Font font) {
    return createBorder(font, null);
}
//
Border border = Utils.createBorder(font);
```

```
Border border = Utils.createBorder(font, null);
```

3. [redacted]

```
String shapeName = Inter.getLocText("FR-Designer_Chart_Circle");
```

```
String shapeName = "";
```

4. [redacted]

```
public String[] getNames4JionTheParty() {
     if (a) {
         return new String[] {"", "", ""};
     } else {
         return new String[0];
     }
}
```

```
public String[] getNames4JionTheParty() {
     if (a) {
         return new String[] {"", "", ""};
     } else {
         return null;
     }
}
```

5. [redacted]if

```
if (shouldDoSomething(a, dim)) {
    doSomething();
}
boolean shuoldDoSomething(int a, Dimension dim) {
    return a > -1 && a != 1 && dim > width && dim.height > 0;
}
```

```
if (a > -1 && a != 1 && dim.width > 0 && dim.height > 0) {
    doSomething();
}
```

6. [redacted]switchcasebreak/return

```
public String diffResult(int type) {
    String someDescription = "abc";
    switch (type) {
        case 1:
          someDescription = "qwe";
          break;
        case 2:
          someDescription = "xyz";
          break;
        default:
            someDescription = "mnx";
            break;
    }
    return someDescription;
}
```

```
public String diffResult(int type) {
    String someDescription = "abc";
    switch (type) {
        case 1:
          someDescription = "qwe";
        case 2:
          someDescription = "xyz";
          break;
        default:
            someDescription = "mnx";
            break;
    }
    return someDescription;
}
```

7. [redacted] hashCode equals :

1) equalshashCode
2) SethashCodeequalsSet
3) MaphashCodeequals

8. [redacted] toArray(T[] array) 0

```
List<String> list = new ArrayList<String>(2);
list.add("guan");
list.add("bao");
array = list.toArray(new String[0]);  // 2018.09.14
```

```
List<String> list = new ArrayList<String>(2);
list.add("guan");
list.add("bao");
array = list.toArray();
```

9. [redacted] foreach remove/add remove Iterator

Iterator -Java

10. [redacted]

```
return booleanExpression;
```

```
if (booleanExpression) {
     return true;
} else {
    return false;
}
```

11. [redacted] protected

12. [redacted] @Override