Clean Code

刘 光聪

liu.guangcong@zte.com.cn

2016.10

内容

命名

驼峰

Identifier	Examples
Class/Struct/Union	Timer, FutureTask, LinkedHashMap, HttpServlet
Method	remove, ensureCapacity, getCrc
Constant	IDLE, ACTIVE, MAX_LINK_NUM
Local Variable	i, xref, houseNumber
Type Parameter	T, E, K, V

反例

```
public interface Run {
  void run();
};
```

正例

```
public interface Runnable {
  void run();
};
```

反例

1. We need to read the password
2. The password has already been read

```
boolean readPassword = true;
```

正例

```
boolean needPassword = true;
boolean userIsAuthenticated = true;
```

反例：记忆包袱

```
ControllerForEfficientHandlingOfStrings
ControllerForEfficientStorageOfStrings
```

反例：作用域相关

```
void onDone() {
  for(int index=0; index!=listeners.size(); index++) {
    listeners[index].onEventDone();
  }
}
```

正例

```
void onDone() {
  for(int i = 0; i != listeners.size(); i++) {
    listeners[i].onEventDone();
  }
}
```

正例

```
void onDone() {
  for(Listener listener : listeners) {
    listener.onEventDone();
  }
}
```

统一语言

1. 设计模式: Factory, Visitor, Repository
2. 工厂方法: valueOf, of, getInstance, newInstance, newType
3. 能力接口: AppendAble, Closeable, Runnable, Readable, Invokable

丰富单词库

Word	Alternatives
send	deliver, dispatch, announce, distribute, route
find	dsearch, extract, locate, recover
start	launch, create, begin, open
make	create, setup, build, generate, compose, add, new

注释

反例

```
int time; // elapsed time in days
```

正例

```
int elapsedTimeInDays;
```

冗余注释

```
/**
 * Constructor
 */
InvokedAtMost(int times);

/**
 * @param Invocation
 * @return boolean
 */
boolean matches(Invocation invocation);
```

常见坏味道

1. 误导性、过时的、与设计实现不匹配的注释
2. 日志型、归属、签名的注释
3. // end if, // end while, // end for, // end try 的注释
4. 已注释掉的代码未删除

必要性

1. 代码无法表达明确的意图时
2. 如果在代码中存在特殊的陷阱或实现方式时

可读性

```
// Fast version of "hash = (65599 * hash) + c"
hash = (hash << 6) + (hash << 16) - hash + c;

// kk::mm::ss, MM dd, yyyy
String timePattern = "\\d{2}:\\d{2}:\\d{2}, \\d{2} \\d{2}, \
  \\d{4}";
```

简化逻辑

反例

```
if (userResult == SUCCESS) {
  if (permissionResult != SUCCESS) {
    reply.writeError(permissionResult);
    return;
  }
  reply.writeError("");
} else {
  reply.writeError(usrResult);
}
```

正例

```
if (userResult != SUCCESS) {
  reply.writeError(usrResult);
  return;
}

if (permissionResult != SUCCESS) {
  reply.writeError(permissionResult);
  return;
}

reply.writeError("");
```

反例

```
if (hour >= 12)  {
  time += "pm";
} else {
  time += "am";
}
```

正例

```
time += (hour >= 12) ? "pm" : "am";
```

反例

```
return exponent >= 0 ? mantissa * (1 << exponent) : mantissa / \
  (1 << -exponent);
```

正例

```
if (exponent >= 0) {
  return mantissa * (1 << exponent);
} else {
  return mantissa / (1 << -exponent);
}
```

反例

```
if (line.split(":")[0].trim().equals("root"))
```

正例

```
String userName = line.split(":")[0].trim();
if ("root".equals(userName)) {
  // ...
}
```

反例

```
if (MAX_STREAM_LEN == length)
```

正例

```
if (length == MAX_STREAM_LEN)
```

英语表达

1. if you are at least 18 years old
2. if 18 years is less than or equal to your age

函数设计

原则

1. 只做一件事，并做好这件事
2. 函数中的每一个语句都在一个相同的抽象层次上

技巧

1. 在同一个抽象层次
2. 命名意图明确
3. 对称性

反例

```
void add(E e) {
  if (!readOnly) {
    int newSize = size + 1;
    if (newSize > elements.length) {
      Object newElements = new Object[elements.length + 10];
      for (int i = 0; i < size; i++) {
        newElements[i] = elements[i];
        elements[i] = null;
      }
      elements = newElements;
    }
    elements[size++] = e;
  }
}
```

正例

```
void add(E e) {
  if (readOnly)
    return;
  if (atCapacity())
    grow();
  addElement(e);
}
```

反例

```
public interface ObjectOutputStream {
  void write(boolean b);
  void write(char c);
  void write(short s);
  void write(int i);
  void write(long l);
  void write(float f);
  void write(double d);
};
```

正例

```
public interface ObjectOutputStream {
  void writeBool(boolean b);
  void writeChar(char c);
  void writeShort(short s);
  void writeInt(int i);
  void writeLong(long l);
  void writeFloat(float f);
  void writeDouble(double d);
};
```

反例

违背 Demeter 法则

```
String outputDir = ctxt.getOptions()
  .getScratchDir()
  .getAbsolutePath();
```

用户接口

```
String outFile = outputDir + File.SEPERATOR + fileName;
FileOutputStream ss = new FileOutputStream(outFile);
```

正例

搬迁职责，隐藏实现

```
FileOutputStream ss = ctxt.readScratchFile(fileName);
```

参考文献

推荐书籍

- Extreme Programming Explained: Embrace Change, 2th, Kent Beck.
- Agile Software Development: Principles, Patterns and Practices, Robert C. Martin.

联系我

- Email: horance@aliyun.com
- Github: https://github.com/horance-liu
- Blog: http://www.jianshu.com/users/49d1f3b7049e

Thanks for Attending