

# 方舟编译器clang-tidy 检查规则解析

PLCT实验室 史宁宁  
2020-03-14

## clang-tidy检查规则

- 1、clang-tidy的检查规则通常会放在项目目录之下的.clang-tidy之中，方舟编译器也不例外；
- 2、规则中以“-”开头的，都是不使用该规则；
- 3、规则中使用的，直接写规则就行。

## clang-tidy检查规则

```
Checks: '-*,readability-identifier-naming,readability-function-size,  
readability-braces-around-statements,readability-magic-numbers,  
misc-unused-parameters,modernize-use-nullptr,modernize-replace-auto-ptr,  
modernize-use-noexcept,modernize-use-override,performance-move-const-arg,  
cppcoreguidelines-pro-type-cstyle-cast,cppcoreguidelines-pro-type-reinterpret-cast,  
cppcoreguidelines-pro-type-const-cast'
```

### CheckOptions:

- key: readability-identifier-naming.ClassCase  
value: CamelCase
- key: readability-identifier-naming.StructCase  
value: CamelCase

## readability-identifier-naming (规则2.1.1)

这条规则是用来检查命名规则的。在这条总的规则之下，还有关于具体要检查项的key和value。比如：

- key: readability-identifier-naming.ClassCase  
value: CamelCase

这里ClassCase说明检查的是类的命名，CamelCase说明要求的风格是大驼峰命名法。方舟里涉及到命名规则，目前有CamelCase（大驼峰命名法）、camelBack（小驼峰命名法）和lower\_case（小写命名法）。涉及到的检查项主要有ClassCase（类）、StructCase（结构体）、TypedefCase（Typedef）、EnumCase（枚举）、EnumConstantCase（枚举常量）、UnionCase（联合）、NamespaceCase（命名空间）、FunctionCase（函数）、VariableCase（变量）、ConstantCase（常量）。

## readability-function-size

这条规则是用来检查函数的大小的。总规则之下，也包含了两条具体的检查项。

- key: readability-function-size.StatementThreshold  
value: 50

检查函数内的语句不超过50条。（建议8.1.1）

- key: readability-function-size.ParameterThreshold  
value: 5

检查函数的参数不超过5个。（建议8.3.3）

## 其他规则

### 3、readability-braces-around-statements

语句必须在大括号之内。这里主要是针对条件判断之后进行执行的语句，比如if-else，哪怕是只有一条语句，也应该在括号之内。（规则3.6.1，规则3.7.1）

### 4、readability-magic-numbers

魔鬼数字检查。（建议9.1.3）

### 5、misc-unused-parameters

未使用函数参数检查。

## 其他规则

### 6、modernize-use-nullptr

检查使用nullptr，而不是NULL或0。（规则10.1.3）

### 7、modernize-replace-auto-ptr

检查禁止使用auto\_ptr。（规则9.5.2）

### 8、modernize-use-noexcept

检查不使用异常机制。（规则11.4.1）

## 其他规则

### 9、modernize-use-override

检查重写虚函数时使用override关键字。(规则10.1.1)

### 10、performance-move-const-arg

检查禁止使用std::move操作const对象。(规则10.1.4)

### 11、cppcoreguidelines-pro-type-cstyle-cast

不使用C风格转换检查。(规则9.3.1)



## 其他规则

12、cppcoreguidelines-pro-type-reinterpret-cast

不使用reinterpret\_cast 检查。（建议9.3.2）

13、cppcoreguidelines-pro-type-const-cast

不使用const\_cast检查。（建议9.3.3）

求关注！ 求四连！