

# Проверка именования в clang-tidy

Программа `clang-tidy` может проверить стилистику регистра идентификаторов, а также их префиксы и суффиксы. Возможные варианты стилей:

```
lower_case
UPPER_CASE
camelBack
CamelCase
camel_Snake_Back
Camel_Snake_Case
aNy_CasE
```

Перечень вариантов идентификаторов приведён [здесь](#).

Для включения проверки именования идентификаторов нужно в находящемся в корне проекта файле `.clang-tidy` включить данный тип диагностики (опция `clang-diagnostic-*,readability-identifier-naming`), а затем указать список выполняемых проверок и их параметры в формате YAML, например:

```
---
Checks:      'clang-diagnostic-*,readability-identifier-naming'
CheckOptions:
  - key: readability-identifier-naming.VariableCase
    value: lower_case
  - key: readability-identifier-naming.VariablePrefix
    value: ''
  - key: readability-identifier-naming.VariableSuffix
    value: ''
---
```

[Проект](#) с файлом `.clang-tidy`, в соответствии с которым проводятся проверки именования идентификаторов.

Перечень возможных параметров конфигурационного файла, их типовые значения и примеры соответствующего кода приведены ниже.

## Синтаксис

### AbstractClass (Абстрактный класс)

Параметр	Значение
AbstractClassCase	CamelCase

Параметр	Значение
AbstractClassPrefix	' '
AbstractClassSuffix	' '

Пример кода до форматирования:

```
class ABSTRACT_CLASS {
    public:
        ABSTRACT_CLASS();
};
```

Пример кода после форматирования:

```
class AbstractClass {
    public:
        AbstractClass();
};
```

## Class (Класс)

Параметр	Значение
ClassCase	CamelCase
ClassPrefix	' '
ClassSuffix	' '

Пример кода до форматирования:

```
class TEST_CLASS {
    public:
        TEST_CLASS();
        ~TEST_CLASS();
};
```

Пример кода после форматирования:

```
class TestClass {
    public:
        TestClass();
        ~TestClass();
};
```

## Struct (Структура)

Параметр	Значение
StructCase	CamelCase
StructPrefix	''
StructSuffix	''

Пример кода до форматирования:

```
struct TEST_struct {  
    int a;  
};
```

Пример кода после форматирования:

```
struct TestStruct {  
    int a;  
};
```

## Union (Объединение)

Параметр	Значение
UnionCase	CamelCase
UnionPrefix	''
UnionSuffix	''

Пример кода до форматирования:

```
union TEST_union {  
    int a;  
    char b;  
};
```

Пример кода после форматирования:

```
union TestUnion {  
    int a;  
    char b;  
};
```

## Enum (Перечисление)

Параметр	Значение
EnumCase	CamelCase
EnumPrefix	''
EnumSuffix	''

Пример кода до форматирования:

```
enum TEST_enum { ONE, TWO };
```

Пример кода после форматирования:

```
enum TestEnum { ONE, TWO };
```

## EnumConstant (Значение в перечислении)

Параметр	Значение
EnumConstantCase	UPPER_CASE
EnumConstantPrefix	''
EnumConstantSuffix	''

Пример кода до форматирования:

```
enum TestEnum { one, TWO };
```

Пример кода после форматирования:

```
enum TestEnum { ONE, TWO };
```

## Namespace (Пространство имён)

Параметр	Значение
NamespaceCase	lower_case
NamespacePrefix	''
NamespaceSuffix	''

Пример кода до форматирования:

```
namespace TEST_ns {  
  
}
```

Пример кода после форматирования:

```
namespace test_ns {  
  
}
```

## InlineNamespace (Вложенное пространство имён)

Параметр	Значение
InlineNamespaceCase	lower_case
InlineNamespacePrefix	''
InlineNamespaceSuffix	''

Пример кода до форматирования:

```
namespace test_ns {  
  inline namespace InlineNamespace {  
  
  }  
} // namespace test_ns
```

Пример кода после форматирования:

```
namespace test_ns {  
  inline namespace inline_namespace {  
  
  }  
} // namespace test_ns
```

## TypeAlias (Псевдоним типа)

Параметр	Значение
TypeAliasCase	lower_case
TypeAliasPrefix	''
TypeAliasSuffix	''

Пример кода до форматирования:

```
struct MyStructure {
    int a;
};
using MY_STRUCT_TYPE = MyStructure;
```

Пример кода после форматирования:

```
struct MyStructure {
    int a;
};
using MyStructType = MyStructure;
```

## Typedef (Объявление типа)

Параметр	Значение
TypedefCase	lower_case
TypedefPrefix	''
TypedefSuffix	''

Пример кода до форматирования:

```
typedef int MY_INT;
```

Пример кода после форматирования:

```
typedef int my_int;
```

## Переменные

### Variable (Переменная) ???

### GlobalVariable (Глобальная переменная)

Параметр	Значение
GlobalVariableCase	lower_case
GlobalVariablePrefix	''
GlobalVariableSuffix	''

Пример кода до форматирования:

```
unsigned GlobalVariable;
```

Пример кода после форматирования:

```
unsigned global_variable;
```

## LocalVariable (Локальная переменная)

Параметр	Значение
LocalVariableCase	camelBack
LocalVariablePrefix	''
LocalVariableSuffix	''

Пример кода до форматирования:

```
int function() {  
    int LocalVariable = 0;  
    return LocalVariable;  
}
```

Пример кода после форматирования:

```
int function() {  
    int localVariable = 0;  
    return localVariable;  
}
```

## StaticVariable (Статическая переменная)

Параметр	Значение
StaticVariableCase	camelBack
StaticVariablePrefix	''
StaticVariableSuffix	''

Пример кода до форматирования:

```
int functionWithStaticVariable() {  
    static int StaticVariable = 0;  
    return StaticVariable;  
}
```

Пример кода после форматирования:

```
int functionWithStaticVariable() {
    static int staticVariable = 0;
    return staticVariable;
}
```

## ConstexprVariable (Переменная constexpr)

Параметр	Значение
ConstexprVariableCase	lower_case
ConstexprVariablePrefix	''
ConstexprVariableSuffix	''

Пример кода до форматирования:

```
constexpr int CONST_FIVE = 5;
```

Пример кода после форматирования:

```
constexpr int const_five = 5;
```

## GlobalPointer (Глобальный указатель)

Параметр	Значение
GlobalPointerCase	lower_case
GlobalPointerPrefix	''
GlobalPointerSuffix	''

Пример кода до форматирования:

```
void* globalPointer;
```

Пример кода после форматирования:

```
void* global_pointer;
```



## LocalPointer (Локальный указатель)

Параметр	Значение
LocalPointerCase	camelBack
LocalPointerPrefix	''
LocalPointerSuffix	''

Пример кода до форматирования:

```
void* function() {  
    void* LocalPointer = nullptr;  
    return LocalPointer;  
}
```

Пример кода после форматирования:

```
void* function() {  
    void* localPointer = nullptr;  
    return localPointer;  
}
```

## Member (Член) ???

### ClassMember (Член класса)

Параметр	Значение
ClassMemberCase	camelBack
ClassMemberPrefix	''
ClassMemberSuffix	''

Пример кода до форматирования:

```
class ClassWithClassMember {  
    public:  
        static int CLASS_MEMBER;  
};
```

Пример кода после форматирования:

```
class ClassWithClassMember {
    public:
        static int classMember;
};
```

## ConstantMember (Константный член)

Параметр	Значение
ConstantMemberCase	lower_case
ConstantMemberPrefix	''
ConstantMemberSuffix	''

Пример кода до форматирования:

```
class ClassWithConstantMember {
    char const ConstMember[4] = "123";
};
```

Пример кода после форматирования:

```
class ClassWithConstantMember {
    char const const_member[4] = "123";
};
```

## PrivateMember (Приватный член)

Параметр	Значение
PrivateMemberCase	camelBack
PrivateMemberPrefix	'm_'
PrivateMemberSuffix	''

Пример кода до форматирования:

```
class ClassWithPrivateMember {
    private:
        int PrivateMember;
};
```

Пример кода после форматирования:

```
class ClassWithPrivateMember {
    private:
        int m_privateMember;
};
```

## ProtectedMember (Защищённый член)

Параметр	Значение
ProtectedMemberCase	camelBack
ProtectedMemberPrefix	''
ProtectedMemberSuffix	''

Пример кода до форматирования:

```
class ClassWithProtectedMember {
    protected:
        int ProtectedMember;
};
```

Пример кода после форматирования:

```
class ClassWithProtectedMember {
    protected:
        int protectedMember;
};
```

## PublicMember (Публичный член)

Параметр	Значение
PublicMemberCase	camelBack
PublicMemberPrefix	''
PublicMemberSuffix	''

Пример кода до форматирования:

```
class ClassWithPublicMember {
    public:
        int PublicMember;
};
```

Пример кода после форматирования:

```
class ClassWithPublicMember {
    public:
        int publicMember;
};
```

## КОНСТАНТЫ

### Constant (Константа) ???

Параметр	Значение
ConstantCase	UPPER_CASE
ConstantPrefix	''
ConstantSuffix	''

Пример кода до форматирования:

```
const int theOne = 1;
```

Пример кода после форматирования:

```
const int THE_ONE = 1;
```

### GlobalConstant (Глобальная константа)

Параметр	Значение
GlobalConstantCase	UPPER_CASE
GlobalConstantPrefix	''
GlobalConstantSuffix	''

Пример кода до форматирования:

```
const int theOne = 1;
```

Пример кода после форматирования:

```
const int THE_ONE = 1;
```

## LocalConstant (Локальная константа)

Параметр	Значение
LocalConstantCase	camelBack
LocalConstantPrefix	''
LocalConstantSuffix	''

Пример кода до форматирования:

```
int functionWithLocalConstant() {  
    const int LocalConstant = 0;  
    return LocalConstant;  
}
```

Пример кода после форматирования:

```
int functionWithLocalConstant() {  
    const int localConstant = 0;  
    return localConstant;  
}
```

## StaticConstant (Статическая константа)

Параметр	Значение
StaticConstantCase	camelBack
StaticConstantPrefix	''
StaticConstantSuffix	''

Пример кода до форматирования:

```
int functionWithStaticConstant() {  
    const int StaticConstant = 0;  
    return StaticConstant;  
}
```

Пример кода после форматирования:

```
int functionWithStaticConstant() {  
    const int staticConstant = 0;  
    return staticConstant;  
}
```

## ClassConstant (Константа класса)

Параметр	Значение
ClassConstantCase	UPPER_CASE
ClassConstantPrefix	''
ClassConstantSuffix	''

Пример кода до форматирования:

```
class ClassWithClassConstant {
    public:
        static int const Class_Constant = 0;
};
```

Пример кода после форматирования:

```
class ClassWithClassConstant {
    public:
        static int const CLASS_CONSTANT = 0;
};
```

## GlobalConstantPointer (Глобальный константный указатель)

Параметр	Значение
GlobalConstantPointerCase	lower_case
GlobalConstantPointerPrefix	''
GlobalConstantPointerSuffix	''

Пример кода до форматирования:

```
void* global_pointer;
void* const GlobalConstPointer = global_pointer;
```

Пример кода после форматирования:

```
void* global_pointer;
void* const global_const_pointer = global_pointer;
```

# LocalConstantPointer (Локальный константный указатель)

Параметр	Значение
LocalConstantPointerCase	camelBack
LocalConstantPointerPrefix	''
LocalConstantPointerSuffix	''

Пример кода до форматирования:

```
void* functionWithLocalConstPointer() {  
    void* pointer = nullptr;  
    void* const LocalConstPointer = pointer;  
    return LocalConstPointer;  
}
```

Пример кода после форматирования:

```
void* functionWithLocalConstPointer() {  
    void* pointer = nullptr;  
    void* const localConstPointer = pointer;  
    return localConstPointer;  
}
```

## Функции

### Function (Функция)

Параметр	Значение
FunctionCase	camelBack
FunctionPrefix	''
FunctionSuffix	''

Пример кода до форматирования:

```
static int static_function() {  
    return 0;  
}
```

Пример кода после форматирования:

```
static int staticFunction() {  
    return 0;  
}
```

## GlobalFunction (Глобальная функция)

Параметр	Значение
GlobalFunctionCase	camelBack
GlobalFunctionPrefix	''
GlobalFunctionSuffix	''

Пример кода до форматирования:

```
int global_function() {  
    return 0;  
}
```

Пример кода после форматирования:

```
int globalFunction() {  
    return 0;  
}
```

## ConstexprFunction (Функция constexpr)

Параметр	Значение
ConstexprFunctionCase	camelBack
ConstexprFunctionPrefix	''
ConstexprFunctionSuffix	''

Пример кода до форматирования:

```
constexpr int get_five() {  
    return ( 5 );  
}
```

Пример кода после форматирования:



```
constexpr int getFive() {
    return ( 5 );
}
```

## Method (Метод) ???

### ClassMethod (Метод класса)

Параметр	Значение
ClassMethodCase	camelBack
ClassMethodPrefix	''
ClassMethodSuffix	''

Пример кода до форматирования:

```
class ClassWithMethod {
    public:
        static int get_int() { return 0; };
};
```

Пример кода после форматирования:

```
class ClassWithMethod {
    public:
        static int getInt() { return 0; };
};
```

### ConstexprMethod (Метод constexpr)

Параметр	Значение
ConstexprMethodCase	camelBack
ConstexprMethodPrefix	''
ConstexprMethodSuffix	''

Пример кода до форматирования:

```
class ClassWithConstexprMethod {
    private:
        constexpr int get_int() { return 0; };
};
```

Пример кода после форматирования:

```
class ClassWithConstexprMethod {
    private:
        constexpr int getInt() { return 0; };
};
```

## VirtualMethod (Виртуальный метод)

Параметр	Значение
VirtualMethodCase	camelBack
VirtualMethodPrefix	''
VirtualMethodSuffix	''

Пример кода до форматирования:

```
class ClassWithVirtualMethod {
    private:
        virtual int get_int() { return 0; };
};
```

Пример кода после форматирования:

```
class ClassWithVirtualMethod {
    private:
        virtual int getInt() { return 0; };
};
```

## PrivateMethod (Приватный метод)

Параметр	Значение
PrivateMethodCase	camelBack
PrivateMethodPrefix	''
PrivateMethodSuffix	''

Пример кода до форматирования:

```
class ClassWithPrivateMethod {
    private:
        int get_int() { return 0; };
};
```

Пример кода после форматирования:

```
class ClassWithPrivateMethod {
    private:
        int getInt() { return 0; };
};
```

## ProtectedMethod (Защищённый метод)

Параметр	Значение
ProtectedMethodCase	camelBack
ProtectedMethodPrefix	''
ProtectedMethodSuffix	''

Пример кода до форматирования:

```
class ClassWithProtectedMethod {
    protected:
        int get_int() { return 0; };
};
```

Пример кода после форматирования:

```
class ClassWithProtectedMethod {
    protected:
        int getInt() { return 0; };
};
```

## PublicMethod (Публичный метод)

Параметр	Значение
PublicMethodCase	camelBack
PublicMethodPrefix	''
PublicMethodSuffix	''

Пример кода до форматирования:

```
class ClassWithPublicMethod {
    public:
        int get_int() { return 0; };
};
```

Пример кода после форматирования:

```
class ClassWithPublicMethod {
    public:
        int getInt() { return 0; };
};
```

## Параметры

### Parameter (Параметр)

Параметр	Значение
ParameterCase	camelBack
ParameterPrefix	''
ParameterSuffix	''

Пример кода до форматирования:

```
int returnInt(int return_Value) {
    return return_Value;
};
```

Пример кода после форматирования:

```
int returnInt(int returnValue) {
    return returnValue;
};
```

### ConstantParameter (Константный параметр)

Параметр	Значение
ConstantParameterCase	camelBack
ConstantParameterPrefix	''
ConstantParameterSuffix	''

Пример кода до форматирования:

```
char returnChar(const char return_Value) {
    return return_Value;
};
```

Пример кода после форматирования:

```
char returnChar(const char returnValue) {  
    return returnValue;  
};
```

## ConstantPointerParameter (Константный указатель на параметр)

Параметр	Значение
ConstantPointerParameterCase	camelBack
ConstantPointerParameterPrefix	''
ConstantPointerParameterSuffix	''

Пример кода до форматирования:

```
void* returnVoid(void* const return_Value) {  
    return return_Value;  
};
```

Пример кода после форматирования:

```
void* returnVoid(void* const returnValue) {  
    return returnValue;  
};
```

## PointerParameter (Параметр типа указатель)

Параметр	Значение
PointerParameterCase	camelBack
PointerParameterPrefix	''
PointerParameterSuffix	''

Пример кода до форматирования:

```
void* returnPtr(void* return_Value) {  
    return return_Value;  
};
```

Пример кода после форматирования:

```
void* returnPtr(void* returnValue) {
    return returnValue;
};
```

## Шаблоны

### TemplateParameter (Параметр шаблона)

Параметр	Значение
TemplateParameterCase	camelBack
TemplateParameterPrefix	''
TemplateParameterSuffix	''

Пример кода до форматирования:

```
template<typename TType> int tFunction(TType t_value) {
    return 0;
}
```

Пример кода после форматирования:

```
template<typename TType> int tFunction(TType tValue) {
    return 0;
}
```

### ParameterPack (Список параметров шаблона)

Параметр	Значение
ParameterPackCase	camelBack
ParameterPackPrefix	''
ParameterPackSuffix	''

Пример кода до форматирования:

```
template<class ...Types> void f(Types ... Parameters_Pack);
```

Пример кода после форматирования:

```
template<class ...Types> void f(Types ... parametersPack);
```

## TemplateTemplateParameter (Параметр шаблона шаблона)

Параметр	Значение
TemplateTemplateParameterCase	camelBack
TemplateTemplateParameterPrefix	''
TemplateTemplateParameterSuffix	''

Пример кода до форматирования:

```
template<template<class T> class TPL_TPL_Parameter>
class AllmightyClass { };
```

Пример кода после форматирования:

```
template<template<class T> class tplTplParameter>
class AllmightyClass { };
```

## TypeTemplateParameter (Типа параметра шаблона)

Параметр	Значение
TypeTemplateParameterCase	CamelCase
TypeTemplateParameterPrefix	''
TypeTemplateParameterSuffix	''

Пример кода до форматирования:

```
template<typename t_type>int tFunction(t_type value) {
    return 0;
};
```

Пример кода после форматирования:

```
template<typename TType>int tFunction(TType value) {
    return 0;
};
```

## ValueTemplateParameter (Значение параметра шаблона)

Параметр	Значение
ValueTemplateParameterCase	camelBack
ValueTemplateParameterPrefix	''
ValueTemplateParameterSuffix	''

Пример кода до форматирования:

```
template<typename TType, int arg_count> int tFunction(TType tValue) {  
    return 0;  
}
```

Пример кода после форматирования:

```
template<typename TType, int argCount> int tFunction(TType tValue) {  
    return 0;  
}
```