C++ Language Syntax

**Outline**

* TGON 엔진에서 사용되는 프로그래밍 언어의 작성 규약을 서술합니다.

**Contents**

1. Global
   1. Function naming
2. Class
   1. Member function naming
3. #include
4. **Global**
   1. **Function naming**
      1. **Create/Make 접두어 함수**
         1. 어떤 객체 또는 POD를 생성하는 함수입니다.

template <typename WindowTy, typename... Args, typename = utility::EnableIfConvertible<WindowTy\*, TWindow\*>>

std::shared\_ptr<WindowTy> MakeWindow(Args&&... args)

{

return std::make\_shared<WindowTy>(std::forward<Args>(args)...);

}

1. **Class**
   1. **Member function naming**
      1. **Initialize 접두어 함수**
         1. 어떤 매개 변수도 제공하지 않고 클래스를 초기화하는 함수입니다.

void AndroidApplication::InitializeConfiguration()

{

m\_config = AConfiguration\_new();

AConfiguration\_fromAssetManager(m\_config, m\_activity->assetManager);

}

* + 1. **Set 접두어 함수**
       1. 어떤 매개 변수를 통해서 클래스를 초기화하는 함수입니다.

void AndroidApplication::SetConfiguration(AConfiguration\*\* config, AAssetManager\* assetManager)

{

\*config = AConfiguration\_new();

AConfiguration\_fromAssetManager(\*config, assetManager);

}

* + 1. **Create/Make 접두어 함수**
       1. 어떤 객체 또는 POD를 생성하는 팩토리 함수입니다.

template <typename WindowTy, typename... Args, typename = utility::EnableIfConvertible<WindowTy\*, TWindow\*>>

std::shared\_ptr<WindowTy> MakeWindow(Args&&... args)

{

return std::make\_shared<WindowTy>(std::forward<Args>(args)...);

}

* 1. Reserved

1. **#include**
   1. include문의 섹션화
      1. 표준 또는 Third-party 라이브러리의 헤더
      2. 현재 헤더와 같은 디렉터리에 존재하는 헤더
      3. 현재 헤더와 다른 디렉터리에 존재하는 헤더

#include <string>

#include <cstdint>

#include <boost/noncopyable.hpp>

#include "GenericWindowType.h"

#include "Core/Platform/TConfig.h"