|  |
| --- |
|  |
| 개발규칙 가이드 |
|  |
|  |
| **Quarter 100’s** |
| **작성자 : 김재환, 서영화, 이효경, 차윤선** |
|  |

|  |
| --- |
|  |

**목차**

[1. 개발 표준 소프트웨어 아키텍처 2](#_Toc77605865)

[1.1 표준 개발규칙 사용 목표 2](#_Toc77605866)

[2. 아키텍처 구성과 패턴 2](#_Toc77605867)

[3. 명명의 정의 2](#_Toc77605868)

[3.1 개발 명명 표기유형 종류 3](#_Toc77605869)

[4. 작명 시 주의사항 4](#_Toc77605870)

[4.1 변수, 메서드, 구조체, 상수 명명 규칙 4](#_Toc77605871)

[5. 개발 규칙 5](#_Toc77605872)

[5.1 들여쓰기 5](#_Toc77605873)

[5.2 한 줄의 최대 길이 5](#_Toc77605874)

[5.3 이항연산자 5](#_Toc77605875)

[5.4 Import 5](#_Toc77605876)

# 개발 표준 소프트웨어 아키텍처

## 표준 개발규칙 사용 목표

표준 개발 규칙을 사용하여, 사용자 요구사항에서부터 개발 완료까지 일련의 프로세스를 자동화 함으로 업무 처리에 있어서 자연스럽게 업무 패턴을 적용함으로써 개발 완료까지 소요되는 시간 및 인력 투자에 대한 비용 절감과 적재 적소에 인력을 조정함으로써 업무의 효율성을 높이는 것을 목표로 합니다. 새로운 시스템 개발에 공용 플랫폼과 표준 프레임 워크를 적용함으로써 유지보수의 편의성을 증대하여 개발 완료된 응용 프로그램의 유지보수 시간을 최대한 줄임으로써 유지보수 비용과 인력을 줄여 비용 절감과 총 소요비용의 절감을 목표로 합니다.

# 아키텍처 구성과 패턴

플랫폼의 가치는 Windows 기반의 플랫폼에서 어느 클라이언트에서도 별도의 개발을 구성하는 것이 아니라 하나의 표준 화면과 하나의 표준 개발 환경으로 Windows 환경 Web환경, 모바일 환경을 지원하는 것을 목표로 합니다.

여기서 "구성"이란 어떻게 하면 목표한 바에 가장 부합하도록 프로젝트를 수행하기 위한 의사결정을 의미합니다. 깔끔하고 효율적인 코드라는 파이썬의 특성을 극대화하는 것을 목표로 합니다. 파이썬의 특성을 극대화 하면서 실행속도에서 다른 컴파일 언어들과의 차이를 줄이기 위해 최적화된 개발을 합니다.

# 명명의 정의

프로젝트 진행 시에 개발자들이 지켜야 하는 몇 가지 규칙을 정합니다. 이 문서에서 정의하는 규칙은 개발자들이 지켜야 하는 필수 규칙을 정의합니다.

이 문서에 정의되어 있는 규칙은 개발 시에 체크인 시에 적용되는 정책으로 개발 전에 미리 숙지해야 합니다. 체크인 정책 적용 시 명명 정의를 지키지 못할 경우 개발 시 문제가 발생할 수 있습니다.

## 3.1 개발 명명 표기유형 종류

3.1.1 Pascal Case

* 각 단어의 첫 문자는 대문자로 시작
* 예제) MyClass

3.1.2 Camel Case

* 첫 문자는 소문자로 시작하여 그 이후의 단어는 대문자로 시작
* 예제) totalCount

3.1.3 Upper Case

* 모든 문자는 대문자로 표시
* 예제) COUNT

3.1.4 Hungarian Notation

* Pascal Case 유형 앞에 변수 타입의 약어를 표시
* 예제) intCount

# 4. 작명 시 주의사항

이 기종간의 언어와 상호 연동 시에 표기법의 혼동을 막기 위하여 대소문자 구분을 엄격히 제한하여 지켜야 하며, 언어의 특성으로 대소문자를 구별할 수 있습니다. 대부분의 프로젝트는 Python으로 이루어지며 대소문자를 구별합니다.

* C/C++, C# : 대소문자 구분
* Basic : 대소문자 구분하지 않음

## 4.1 변수, 메서드, 구조체, 상수 명명 규칙

프로젝트에서 변수 명, 클래스 명, 메서드 명등을 정의하기 위하여 다음과 같은 규칙을 지켜야 합니다.

변수 명 : Camel Case

* totalCount

메서드 명 : Pascal Case

* MyMethod()

구조체 명: Upper Case

* MyStruct

상수 명: Upper Case

* MyCONST = 1

컴포넌트명: 대소문자를 구분하지 않음

예제) System.IO

대소문자로 구분되는 파라미터, 속성, 메서드는 사용하지 않습니다.

예제) private int UserLogin(string userID, userPassword)

int ThreadTotal{ get ; set ;}

void UserLoginSetting {}

명명 시에는 .NET Framework 에서 사용되는 클래스 명, 메서드, 내장 함수 명을 사용하지 않습니다. 명명 시에는 “동사”+”명사” 또는 “명사” + “명사”를 기본으로 합니다.

# 5. 개발 규칙

## 5.1 들여쓰기

파이썬에서 코드블록은 스코프({})가 아닌, 들여쓰기로 나타냅니다.

들여쓰기 당 4개의 space를 사용합니다.

괄호(대,중,소) 및 괄호안의 괄호와 같이 연결되는 라인에서 줄바꿈이 일어나는 요소들은 수직으로 정렬되어야 합니다.

python3의 경우 탭과 스페이스의 조합은 문법적 오류를 야기합니다.

## 5.2 한 줄의 최대 길이

닥스트링이나 주석과 같이 구조적 제한이 적은 긴 텍스트 블럭을 처리하기 위해선, 한 줄의 최대 길이는 72 문자로 제한되어야 합니다.

## 5.3 이항연산자

연산자는 스크린 상 여러 열로 흩어지는 경향이 있기 때문에 항상 이항연산자 전에 줄바꿈을 합니다.

## 5.4 Import

Import는 항상 파일의 가장 위쪽에 모듈의 주석과 닥스트링 뒤에, 모듈 전역변수나 상수 전에 위치시키도록 합니다.