|  |
| --- |
|  |
| 개발규칙 가이드 |
|  |
|  |
| **우원하, 김민지, 조상우, 강경문** |
|  |
|  |

|  |
| --- |
|  |

목차

[1. 개발 표준 소프트웨어 아키텍처 2](#_Toc77577297)

[1.1 화면 및 UI 구성 2](#_Toc77577298)

[1.2 표준 개발규칙 사용 목표 2](#_Toc77577300)

[2. 아키텍처 구성과 패턴 2](#_Toc77577301)

[3. 명명의 정의 3](#_Toc77577302)

[3.1 개발 명명 표기유형 종류 3](#_Toc77577303)

[4. 작명 시 주의사항 5](#_Toc77577304)

[4.1 변수, 클래스, 메서드 명명 규칙 5](#_Toc77577305)

[5. 네임스페이스 6](#_Toc77577306)

[6. 클래스의 정의 6](#_Toc77577310)

[6.1 클래스 접근권한 6](#_Toc77577322)

[7. 공통 명명규칙 7](#_Toc77577324)

[7.1 명명 표기법 기본 7](#_Toc77577330)

# 개발 표준 소프트웨어 아키텍처

## 화면 및 UI 구성

본 개발에서 Python 언어의 역할은 응용 프로그램 만들 수 있는 플랫폼을 제공합니다. 이 플랫폼을 기반으로 데이터를 크롤링, 정제 후 그래프를 출력하게 되므로, 문제 없는 소프트웨어를 개발하는 것을 목표로 아키텍처를 구성합니다.

소프트웨어 아키텍처는 시스템 구조들로 각 구조의 역할 부분에 소프트웨어 컴포넌트들을 구성하여 화면에 표시되는 속성들과 내부 구조들 간의 관계로 이루어 졌습니다.

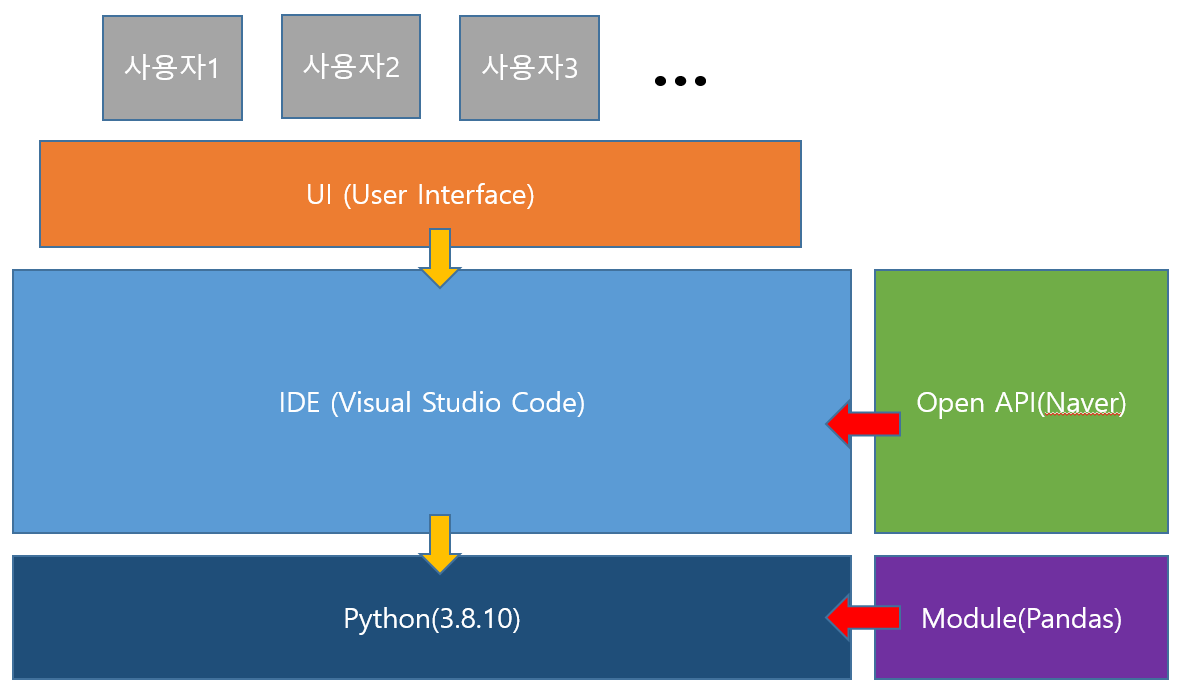
화면에 그래프와 같은 시각화 자료를 출력하기 위해서 Pandas 모듈을 이용할 예정이고, 해당 모듈을 기반으로 화면에 표시될 항목들을 구성합니다.

## 표준 개발규칙 사용 목표

프로젝트 개발 시, 개발 생산성의 향상 그리고 운영 및 유지보수의 효율화를 위해서 반드시 준수되어야 할 항목들로 구성하여 표준 개발규칙 준수사항을 정의합니다.

# 아키텍처 구성과 패턴

저희가 제작할 프로그램의 아키텍처는 다음과 같이 구성되어 있습니다.



Python을 사용하여 Visual Studio Code을 통합 개발 환경 툴로 선정하여 이용할 예정이고, Python 에서는 Pandas 모듈을 사용하여 보다 정교하게 시각화하여 그래프 형태로 출력되고, 오픈 API 중 네이버의 데이터랩스 관련 API를 사용할 예정입니다. 그리고 이러한 구성요소들이 통합되어 UI가 만들어지고, 사용자는 해당 UI를 통해 프로그램을 이용할 수 있습니다.

# 명명의 정의

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

개발을 위해 구성된 팀원들의 보다 효율적인 협업을 위해서는 사전에 미리 합의된 명명 규칙을 이용하는 것이 유리합니다.

해당 명명 규칙들은 숙지해야 합니다. 체크인 정책 적용 시 명명 정의를 지키지 못할 경우 개발 시 문제가 발생할 수 있습니다.

## 개발 명명 표기유형 종류

1. Pascal Case

* 각 간어의 첫 문자는 대문자로 시작
* 예제) MyClass

1. Camel Case

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

1. Upper Case

* 모든 문자는 대문자로 표시
* 예제) COUNT

1. 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. 대 부분의 프로젝트는 Python으로 이루어지며 대소문자를 구별합니다.

* Python : 대소문자 구분

## 4.1 변수, 클래스, 메서드 명명 규칙

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

1. 변수 명 : Camel Case
   * totalCount
2. 메서드 명 : Pascal Case
   * MyMethod()
3. 클래스 명: Pascal Case
   * MyClass()
4. 상수 명: Upper Case
   * MyCONST = 1
5. 컴포넌트명: 대소문자를 구분하지 않음

예제) System.IO

명명 시에는 Python에서 사전 예약된 키워드를 쓰지 않습니다. (에러 발생)

명명 시에는 “동사”+”명사” 또는 “명사” + “명사”를 기본으로 합니다.

# 5. 네임스페이스

* 네임스페이스를 정의할 때에는 모호성을 방지하여야 합니다.
* 참조 단순화를 구현하기 위하여 네임스페이스는 10자이상을 넘지 않도록 정의하고 10자를 넘길 경우 협의 후에 약어를 이용하여 10자 이내로 작명합니다.
* **Built-In 네임스페이스 :**파이썬에서는 'Built-In’ 네임스페이스라는 최상위 네임스페이스가 있습니다. 이는 마치 윈도우즈 파일 시스템에서 루트 폴더와 유사합니다.
* **Global 네임스페이스 :**  
  그 아래에는 모듈마다 '전역 네임스페이스'라는 것이 존재하게 됩니다. 그 중 하나가 우리가 파이썬 실행 후 shell에서 명령을 바로 내릴 때 자동적으로 위치하게 되는 '\_\_main\_\_'라는 이름의 네임스페이스입니다.
* **Local 네임스페이스 :**  
  전역 네임스페이스 내에 존재하는 함수 등 객체 내부를 지칭하는 지역 네임스페이스들이 있습니다. locals() 함수를 실행하면 이 함수를 실행하는 공간에 해당하는 네임스페이스에 존재하는 객체들과 그 식별자들을 볼 수 있습니다.

# 6. 클래스의 정의



Python에서 사용되는 클래스에 대한 내용을 정의합니다.

클래스의 구성 항목들은 아래와 같습니다.

* 메서드
* 속성



## 6.1 클래스 접근권한

* Public : 제한 없음
* Protected : 파생된 클래스까지
* Private : 현재 정의된 곳 내
* default: 같은 클래스 내부와 같은 패키지 내

# 7. 공통 명명규칙



## 명명 표기법 기본

|  |  |  |
| --- | --- | --- |
| **식별자** | **유형** | **용례** |
| 네임스페이스(Namespace) | User Define | ”<팀명>” |
| 프로젝트(Project) | User Define | ”<팀명>”. Project |
| 모듈(import 모듈) | Pascal | import + 모듈명 |
| 클래스(Class) | Pascal | MessageHandler |
| 예외 클래스(Exception Class) | Pascal | HandleException  항상 Exception으로 종료 |
| 인터페이스(Interface) | Pascal | IDisposasble  항상I로 시작 |
| 메서드(Method) | Pascal | ToString |
| 프로퍼티(Property) | Pascal | BackColor |
| 변수(Variable) | Camel | documentLibaray |
| 상수(Constant) | Upper Case | MAX\_VALUE |
| 파라미터(Parameter) | Camel | typeName |
| 읽기전용 필드 | Pascal | RedValue |