

#5 코딩 표준(변수 명명법)

2020년 12월 1일 화요일 오전 10:14

Naming Rule

1. PascalCasing (파스칼 케이싱)

- 클래스, 열거형, 이벤트, 메서드 등의 이름을 만들 때에는 대문자로 시작하는 변수명을 사용한다.
- 복합어일 경우 중간에 시작하는 새로운 단어는 대문자로 적는다.

예) `UtilityBox`, `MainFrame`

2. CamelCasing (카멜 케이싱)

- 메서드의 매개변수의 이름에 적용되는데 첫번째 문자는 소문자로 시작하고 복합어 일 경우 파스칼 케이싱과 동일하게 적용한다.
- 동일한 이름을 가지는 두 항목을 구분하는 용도로도 사용한다.

예) `utilityBox`, `mainFrame`

3. GNU Naming Convention

- Linux의 프로젝트들은 GNU Naming Convention이라는 형태의 명명법을 주로 사용한다.
- 모두 소문자를 사용하고 복합어 사이를 '_'를 사용하여 연결한다.

예) `gtk_widget_activate`

4. Hungarian notation (헝가리안 표기법)

- Microsoft 의 개발자중 헝가리 사람의 프로그래머가 쓰던 변수 명명법으로 MS내부에서 따라쓰기 시작하던 것이 점차 전세계의 프로그래머들에게 널리 퍼져 이젠 프로그램 코딩 시 변수 명명의 표준적인 관례가 되었다.
- C#에서는 이러한 명명법을 사용하지 않고 있으며 주로 윈도우즈 프로그래밍에 사용된다.

예) `g_bTrue`

- 첫글자 g는 전역변수, m은 멤버변수를 의미한다. 전역이나 멤버변수의 경우에는 그 다음에 _를 적는다.
- b는 Boolean타입을 의미하고 True가 의미있는 이름이다.

예) `nCnt`

- 전역이나 멤버변수가 아니므로 g_ 나 m_ 가 없다.
- n과 i는 자연수를 뜻하며 i는 주로 인덱스에 사용하고, n은 카운트의 목적에 주로 사용한다.
- 의미있는 이름이 길 경우에는 자음만을 사용한다.

5. BREW Naming Convention

- BREW 는 Qualcomm에서 만든 플랫폼으로 국내의 휴대폰 제조사들은 초기부터 현재까지 이 코드들을 많이 사용하고 있다.
- 기존 명명법을 조합한 형태로 변종 명명법이지만, 익숙함을 벗어나지 못하는 국내 제조

사의 개발자들이 선호하는 형태이다.

- 클래스나 인터페이스를 대문자나 파스칼 케이싱으로 앞에 두고, '_' 이후에 다시 파스칼 케이싱 형태의 메서드 명을 적는다.

예) `IDISPLAY_ClearScreen`

6. Constant (상수)

- 거의 모든 명명법에서 상수를 표기하는 방법은 거의 동일하다.
- 모든 문자를 대문자로 사용하는 GNU Naming Convention의 형태를 사용한다.

예) `DEFAULT_COUNTRY_CODE`

다양한 명명법들이 존재한다. 어떤 방법이 가장 좋다는 것은 없다. 프로젝트의 상황에 가장 적합한 명명법을 팀에서 결정하는 것이다. 프로젝트의 초기에 명명법을 결정하고 모든 개발자들이 규칙을 따라 코드를 작성하는 것이다.