

변수명/함수명 규칙 (2023.05.20. ver)

작성자 : 20191033 황현석

코드가 길어짐에 따라 추후 코드의 의미를 파악하기 어려워지는 것을 방지하고자 변수명 규칙을 짓고자 한다.

[변수명]

1. 변수명을 보고 생각하도록 하는 변수명은 지양한다. 지나치게 줄여쓴 변수명도 지양한다. 단, 통용적으로 사용하는 줄임말인 경우 줄임말을 인정한다. (Database -> DB)

Ex. 카드 변수 => card (O) , ca (X)

Ex. 데이터베이스 => DB (O), database (X)

2. 최대한 **전처리기를 활용한 상수**를 정의하자.

flags |= 0x10; 은 로드가 되었음을 설정하는 코드라고 했을 때, 0x10을 그냥 사용하는 것보다 #define 0x10 LOADED_FLAG;처럼 전처리기로 정의하고 flags |= LOADED_FLAG; 구문을 사용하는 것이 의미를 파악하기 더 쉽다.

전처리기를 사용한 상수 정의의 경우 해당 프로젝트에서 대문자로 사용하는 것을 원칙으로 한다.

3. 배열, 리스트의 경우 각 맞는 표시를 붙일 것

Ex) card_arr - 카드 배열을 의미하는 변수명

4. 변수 대소문자 기준

<모두 대문자>

- 상수
- 관습적으로 자주 대문자로 사용하는 것 (ID, DB, PW, IP)

<첫글자 대문자 + 이어지는 단어 대문자>

- 인터페이스, 클래스 이름

<첫글자 소문자 + 이어지는 단어를 대문자>

- 일반 변수/함수

[공통]

1. 카멜 표기법을 사용한다.

2-1. 스코프가 짧은 경우 (지역변수 정도) 각 루프에 맞는 관례적인 변수를 사용한다.

Ex. for루프에서 관례적으로 반복 변수를 사용할 때 변수명 **i**를 사용한다.

2-2. 스코프가 긴 경우 (전역변수, 클래스 멤버 변수 및 함수) 최대한 명시적으로 이름을 선택한다.

[함수명]

3. 함수의 이름을 지을 때 **함수가 하는 일이 명확하고 자세하게**(상세하게) 나타나도록 짓는다.

Ex. 카드 분배하는 역할을 하는 함수 => CardDistribute

단, 한 가지의 함수는 **한 가지의 일만** 하도록 모듈화한다. - 단일 책임 원칙 준수

Ex. 한 함수에서 카드 분배도 하고, 카드 내기도 하는 함수 => X

Ex. 한 함수에서 카드 분배만 진행하는 함수 => O

만약 한 가지의 일이 하도록 함수가 정의되어 있지 않은 경우 함수를 분리하여 한 가지의 함수에 한 가지의 일(기능)만 할 수 있도록 해야 한다. [매우 중요]

사례 예시)

변수명을 생각하다가, 하나의 함수에 한 가지의 일(기능)을 하지 않는 경우 자연스럽게 문제가 발생할 것이다.

(ex. 카드를 분배하여 분배한 카드의 결과값을 출력하는 함수

=> CardDistributeAndPrintPlayerCards)

변수명을 생각할 때 이렇게 And가 떠오른다면, 모듈화를 더 해야 된다는 것이다.

다음과 같이 모듈화를 더 할 수 있다.

각 플레이어에게 카드를 분배하는 역할을 하는 함수 => CardDistribute

플레이어가 분배한 카드를 출력하는 함수 => PrintPlayerCards