

Day-19

Enumerator

열거형

enum(열거형)은 명명된 정수 상수 집합을 그룹화하는 데이터 유형

```
enum color_t {BLACK, BLUE, GREEN};

color_t color = BLUE;
cout << (color == BLAKC); // print false
```

문제

```
enum color_t {BLACK, BLUE, GREEN};
enum fruit_t {APPLE, CHERRY};

color_t color = BLACK;           // int : 0
fruit_t fruit = APPLE;           // int : 0
bool b = (color == fruit);       // print 'true'
```

Strongly Typed Enumerator - enum class

열거형 클래스(범위 열거형) 데이터 타입은 암시적으로 int로 변환할 수 없는 타입 안전 열거형임

```
enum class Color { BLACK, BLUE, GREEN };
enum class Fruit { APPLE, CHERRY };
Color color = Color::BLUE;
Fruit fruit = Fruit::APPLE;
```

```
// bool b = (color == fruit)
// 위 코드는 컴파일 에러, 과일과 색상을 일치 시키려고 하지만 둘은 완전히 다른 타입
// int a1 = Color::GREEN; 컴파일 에러
// int a2 = Color::RED + Color::GREEN; 컴파일 에러
int a3 = (int) Color::GREEN; // 명시적 형변환을 적용하면 가능함
```

enum, enum class 특징

- enum, enum class는 비교할 수 있음

```
enum class Color { RED, GREEN, BLUE };
cout << (Color::RED < Color::GREEN); // print true
```

- enum, enum class는 증가하는 순서대로 자동으로 열거됨

```
enum class Color { RED, GREEN = -1, BLUE, BLACK };
//           (0)   (-1)           (0)   (1)
Color::RED == Color::BLUE; // true
```

- enum, enum class는 별칭을 포함할 수 있음

```
enum class Device { PC = 0, COMPUTER = 0, PRINTER };
```

- C++11 enum, enum class로 기본 유형 설정 가능

```
enum class Color : int8_t { RED, GREEN, BLUE };
```

enum class 특징 - C++17

- C++17 enum class는 직접 목록 초기화를 지원

```
enum class Color { RED, GREEN, BLUE };
Color a{2}; // ok, Color::BLUE와 같음
```

- C++17 enum, enum class는 attributes(특성)을 지원함

- attributes : C++11~ 특성(attribute)이 추가되어 컴파일러에게 부가 정보를 전달하는 방식을 표준화함

```
enum class Color { RED, GREEN, BLUE [[deprecated]] };
auto x = Color::BLUE; // compiler warning
```

- deprecated : 소멸 예정인 것을 컴파일 경고로 알려줌

enum class 특징 - C++20

- C++20에서는 열거자 식별자를 로컴 범위에 도입하여 복잡성을 줄일 수 있음

```
enum class Color { RED, GREEN, BLUE };
switch (x) {
    using enum Color; // C++20
    case RED:
    case GREEN:
    case BLUE:
}
```

enum, enum class - 일반적인 오류

- enum, enum class는 항상 초기화되어야 함
- C++17 enum, enum class의 기본 유형과 관련하여 범위를 벗어난 값에서 캐스팅하면 정의되지 않은 동작이 발생함

enum, enum class 와 constexpr

- C++17 constexpr 표현식은 명시적 기본 타입이 없는 열거형에 대해 범위를 벗어난 값을 허용하지 않음

```
enum Color { RED }; // 명시적인 기본 타입이 없는 열거형
enum Fruit : int { APPLE }; // int가 명시적 기본 타입임
enum class Device { PC };
// constexpr Color a1 = (Color) -1; 컴파일 에러
```

```
const Color a2 = (Color) -1; // ok  
constexpr Fruit a3 = (Fruit) -1; // ok  
constexpr Device a4 = (Device) -1; // ok
```