Basic_Concepts_3 - constexpr

constexpr

constexpr 문은 컴파일 타임에 평가할 수 있다는 표현을 선언한다.

- `const` 변수의 값이 프로그램의 실행 전반에 변하지 않을것임을 보장한다.
- `constexpr`은 `const`를 나타낸다.
- `constexpr`은 성능과 메모리 사용을 돕는다.
- `constexpr`은 컴파일타임에 영향을 줄 가능성이 있다.

`constexpr` 변수는 언제나 컴파일 타임에 평가된다.

```
const int v1 = 3; // 컴파일 타임 평가
const int v2 = v1 * 2; // 컴파일 타임 평가

int a = 3; // "a" 는 동적이다.
const int v3 = a; // 런타임 평가

constexpr int c1 = v1; // ok
// constexpr int c2 = v3; // 컴파일 에러. v3은 동적이다.
```

constexpr 함수

constexpr 함수는 매개변수가 컴파일타임에 평가되는 한 컴파일 타임에 평가됨을 보장한다.

- `constexpr`로 이름지어지지 않은 런타임 함수는 포함할 수 없다.
- C++11: 반드시 하나의 return 상태만 속해야 하며, 반복문 혹은 스위치는 필히 속하면 안된다.
- C++14: 제한이 없다.

```
constexpr int square(int value) {
    return value * value;
}
```

```
square(4); // 컴파일 타임 평가
int a = 4; // "a"는 동적이다.
square(a); // 런타임 평가
```

constexpr의 한계

- try-catch blocks, exceptions 그리고 RTTI처럼 런타임 특징은 포함할 수 없다.
- `goto`와 `asm` 상태는 포함할 수 없다.
- `static` 변수는 포함할 수 없다.
- `assert()`는 C++14까진 포함할 수 없다.
- C++20까진 `virtual`을 사용할 수 없다.
- 미 정의 행동 코드를 허가하지 않는다.
- `constrexprt` 함수는 런타임 객체의 경우 static 멤버가 아니면 사용할 수 없다.
- `static constexpr` 멤버 함수는 특정 인스턴스에 의존하지 않기 때문에 문제가 생기지 않는다.