Day-27

Initialization

Variable Initialization, Uniform Initialization, Brace Initialization Advantages

Variable Initialization

C++03

```
int a1;             // default 초기화(정의되지 않은 값)

int a2(2);          // 직접(또는 값) 초기화
int a3(0);          // 직접(또는 값) 초기화(제로 초기화)
// int a4();        // a4는 함수

int a5 = 2;         // 복사 초기화
itn a6 = 2u;        // 복사 초기화 (+ 암시적 변환)
int a7 = int(2);    // 복사 초기화
int a8 = int();     // 복사 초기화 (제로 초기화)

int a9 = {2};       // 복사 목록 초기화
```

Uniform Initialization

중괄호 초기화 또는 중괄호 초기화 목록이라고도 하는 C++11 **유니폼 초기화** 구문을 사용하면 다양한 엔티티(변수, 객체, 구조체 등)를 일관된 방식으로 초기화 가능

```
int b1{2};          // 직접 목록(또는 값) 초기화
int b2{};           // 직적 목록(또는 값) 초기화 (제로 초기화)

int b3 = int{};     // 복사 초기화 (제로 초기화)
```

```
int b4 = int{4};  // 복사 초기화

int b5 = {};      // 복사 목록 초기화 (제로 초기화)
```

Brace Initialization Advantages

유니폼 초기화는 산술 타입을 안전하게 변환하는데도 사용할 수 있음,
즉 잠재적인 값 손실을 방지할 수 있음
구문도 최신 형 변환보다 더 간결함

```
int b4 = -1;          // ok
int b5{-1};           // ok
unsigned b6 = -1;     // ok
// unsigned b7{-1};   // 컴파일 에러

float f1{10e30};      // ok
float f2 = 10e40;     // ok, "inf" 값
// float f3{10e40};   // 컴파일 에러
```