

🏠 / C++프로그래밍과실습 (CB3500572-062) / 실습 074 - cast operator (type conversion) / 개요

개요

제출

편집

코딩 결과

실습 074 - cast operator (type conversion)

제출 마감일: 2023-05-07 23:59

업로드 가능한 파일 수: 3

제출 방식: 개인

목적

이 실습은 class 의 cast operator 를 연습합니다.

설명

형 변환 (typecasting) 은 하나의 데이터 타입을 다른 데이터 타입으로 변경하는 것을 의미합니다.

예를 들어, double 타입의 변수를 int 타입으로 변경하는 것입니다.

```
double pi = 3.14;
int iPI = pi; // implicit conversion (묵시적 형변환)
std::cout << iPI << std::endl;
```

명시적으로 형 변환 할 수도 있습니다.

```
std::cout << (int) pi << std::endl;
```

출력은 예상했던 대로 3 으로 동일합니다.

객체 역시 형 변환 할 수 있습니다.

어디 쓰냐구요? 벌써 사용해 본 분들도 있습니다.

실습 5-6 Builder 클래스 만들기에서 은근 슬쩍 사용 되었습니다.

알림 객체의 정의를 살펴보면, create() 정적 멤버 함수의 반환 값이 NotificationBuilder 타입입니다.

```
class Notification {
public:
    static NotificationBuilder create();
    ...
}
```

그런데 알람 객체를 생성하는 코드를 살펴보면 좌측이 NotificationBuilder 타입이 아니라 Notification 타입이라는 것을 알 수 있습니다. (묵시적 형변환)

```
Notification notification = Notification::create()
    .setSmallIcon(1)
    .setContentTitle("Justin Rhyss")
    .setContentText("Do you want to go see a movie tonight?")
    .setPriority(5);
```

또 다른 활용 예를 살펴봅시다.

현실의 문제에서 돈을 다루는 프로그램을 작성하는 경우가 많습니다.

영화 예매, 쇼핑물, 게임 등의 앱을 사용해 보면 결제해야 하는 경우가 생깁니다.

이런 도메인에서 결제 관련 프로그램을 개발하는 개발자는 돈을 어떤 타입으로 내부적으로 다루는지 고민이 될 수 밖에 없습니다.

마치 std::string 타입처럼, 자신이 만들 프로그램 여기 저기에 영향을 안 미치는 곳이 없을 정도로 돈 객체는 중요할 것 같습니다.

그럼 원(₩) 을 객체로 한번 표현해 봅시다.

```
class Won {
public:
    Won() = default;
    Won(long amount) : amount(amount) {}
    operator long () const {
        return amount;
    }
private:
    long amount=0;
};
```

자 한번 사용해 봅시다!

```
Won won5000{5000};
Won won10000{10000};
std::cout << won5000 + won10000 << std::endl;
```

컴파일이나 될까요?

컴파일러에게 명시적 형 변환을 강제하고 싶다면 explicit 키워드를 이용해 봅시다.

```
explicit operator long () const {
    return amount;
}
```

이제 묵시적 형 변환은 안되고, 명시적으로 형 변환 해줘야 합니다.

```
std::cout << (long) won << std::endl;
```

문제

Dollars 객체와 Cents 객체를 생성하는 클래스를 정의하시오.
단, 달러 객체는 센트 객체로 형 변환 가능하며,
센트 객체는 int 타입으로 명시적 (explicit) 으로 형 변환되도록 구현하시오.

<참고>

```
// DollarsTest.cpp -----
```

```
#include <iostream>
#include "Dollars.h"
#include "Cents.h"

int main() {
    Dollars dollar {1};
    Cents cent = (Cents) dollar;
    std::cout << (int) cent << std::endl;
}
```

입력

없음

출력

100

제출파일

Dollars.h
Cents.h
74.csv