

윈도우 프로그래밍

4. 클래스(1)

2018. 3.30.
심미나 교수



목 차

I. C++의 구조체

II. 실습

I. C++의 구조체



구조체의 활용

- 서로 관련된 데이터를 하나로 묶어 프로그램 구현/관리의 편의성 증가
 - 관련된 데이터들은 생성 및 소멸 시점, 이동 및 전달 시점 또는 방법이 일치하므로
- 즉, 구조체란, 연관된 데이터를 하나로 그룹화하는 문법적 장치

```
(형식) struct 구조체명 {  
    자료형  멤버변수;  
    자료형  멤버변수;  
    .....  
    .....  
};
```

```
struct Employee  
{  
    char name[20];  
    char dept[30];  
    char job[30];  
    int  year;  
    char tel[20];  
    char email[40];  
};
```



- 이름
- 소속
- 담당업무
- 재직기간
- 전화번호
- 메일주소

C++의 구조체



구조체 변수의 선언

- 구조체 변수 선언

(형식) 구조체명 구조체 변수명;

- C와 달리 struct 키워드 생략 위한 typedef선언 불필요

```
struct Car basicCar;  
struct Car simpleCar;
```



```
Car basicCar;  
Car simpleCar;
```

```
struct Car  
{  
    char gamerID[ID_LEN]; // 소유자ID  
    int fuelGauge;         // 연료량  
    int curSpeed;          // 현재속도  
};
```



- 소유주
- 연료량
- 현재속도
- 취득점수
- 취득아이템

C++의 구조체



구조체 변수의 선언

- 구조체 변수 선언

(형식) 구조체명 구조체 변수명;

- 데이터 뿐만 아니라, 해당 데이터와 연관된 함수도 하나로 그룹화하여 활용 가치를 부여

```
struct Car
{
    char gamerID[ID_LEN]; // 소유자ID
    int fuelGauge;         // 연료량
    int curSpeed;          // 현재속도
};
```

```
void ShowCarState(const Car &car)
{
    . . . .
}

void Accel(Car &car)
{
    . . . .
}

void Break(Car &car)
{
    . . . .
}
```

C++의 구조체



구조체 변수의 선언

• 구조체 내 함수 삽입

- C++에서는 구조체 내에 함수 삽입이 가능 → ‘클래스’
- 구조체 내에 함께 선언된 변수에는 직접 접근이 가능

```
void ShowCarState()
{
    cout<<"소유자ID: "<<gamerID<<endl;    // 위에 선언된 gamerID에 접근
    cout<<"연료량: "<<fuelGauge<<"%"<<endl;
    cout<<"현재속도: "<<curSpeed<<"km/s"<<endl<<endl;
}
```

```
void Break()
{
    if(curSpeed<BRK_STEP)
    {
        curSpeed=0;    // 위에 선언된 curSpeed에 접근
        return;
    }

    curSpeed-=BRK_STEP;
}
```

```
struct Car
{
    char gamerID[ID_LEN];
    int fuelGauge;
    int curSpeed;

    void ShowCarState()
    {
        . . . .
    }

    void Accel()
    {
        . . . .
    }

    void Break()
    {
        . . . .
    }
};
```

C++의 구조체



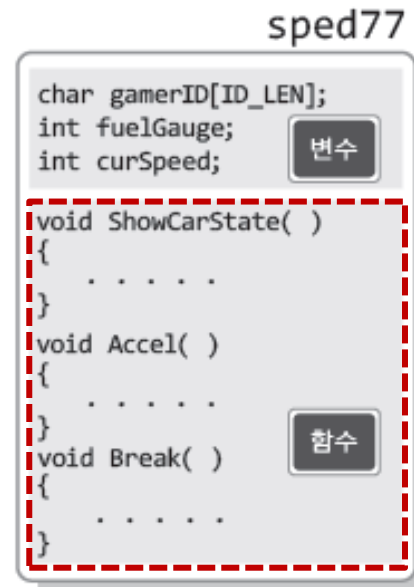
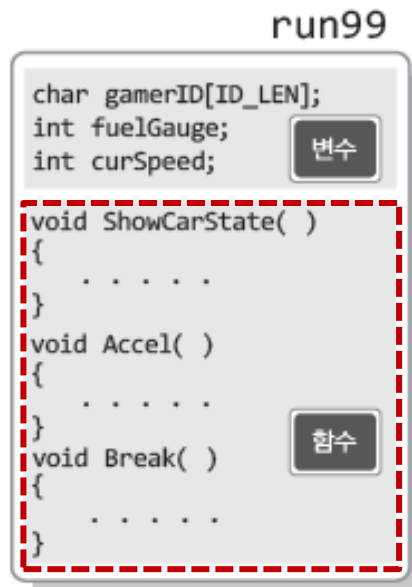
구조체 변수의 선언

• 구조체 변수의 생성

- 생성된 구조체 변수마다 함수가 독립적으로 존재하는 구조로 보임(논리적 의미)
→ ‘변수(객체)’로 이해하기!

(예시)

- Car run99 = {“run99”, 100, 0}
Car sped77 = {“sped77”, 100, 0}



C++의 구조체



구조체 정의

- 구조체 내에 enum 상수의 선언

- 구조체 내에 enum 선언을 함으로써 잘못된 외부 접근을 제한 가능
- (예시) Car 클래스를 위해 정의된 상수

```
namespace CAR_CONST
{
    enum
    {
        ID_LEN      =20,
        MAX_SPD      =200,
        FUEL_STEP     =2,
        ACC_STEP      =10,
        BRK_STEP       =10
    };
}
```

```
#define ID_LEN      20
#define MAX_SPD     200
#define FUEL_STEP    2
#define ACC_STEP     10
#define BRK_STEP     10
```

* 연관된 상수들을 하나의
이름공간에 별도로 그룹화 가능

```
struct Car
{
    enum
    {
        ID_LEN      =20,
        MAX_SPD      =200,
        FUEL_STEP     =2,
        ACC_STEP      =10,
        BRK_STEP       =10
    };

    char gamerID[ID_LEN];
    int fuelGauge;
    int curSpeed;

    void ShowCarState() { . . . . }
    void Accel() { . . . . }
    void Break() { . . . . }
};
```

C++의 구조체



구조체 정의

- 구조체 내에 선언된 함수의 정의

- 구조체 안에 정의된 함수는 inline 선언된 것으로 간주

```
struct Car
{
    . . . . .
    void ShowCarState();
    void Accel();
    . . . . .
};
```

① 구조체 내 삽입된 함수 선언

```
void Car::ShowCarState()
{
    . . . . .
}

void Car::Accel()
{
    . . . . .
}
```

① 구조체 내 선언된 함수 정의

- 외부에 함수 정의시 명시적 inline 선언

```
inline void Car::ShowCarState() { . . . . . }
inline void Car::Accel() { . . . . . }
inline void Car::Break() { . . . . . }
```

IV. 실습



감사합니다

mnshim@sungkyul.ac.kr

