



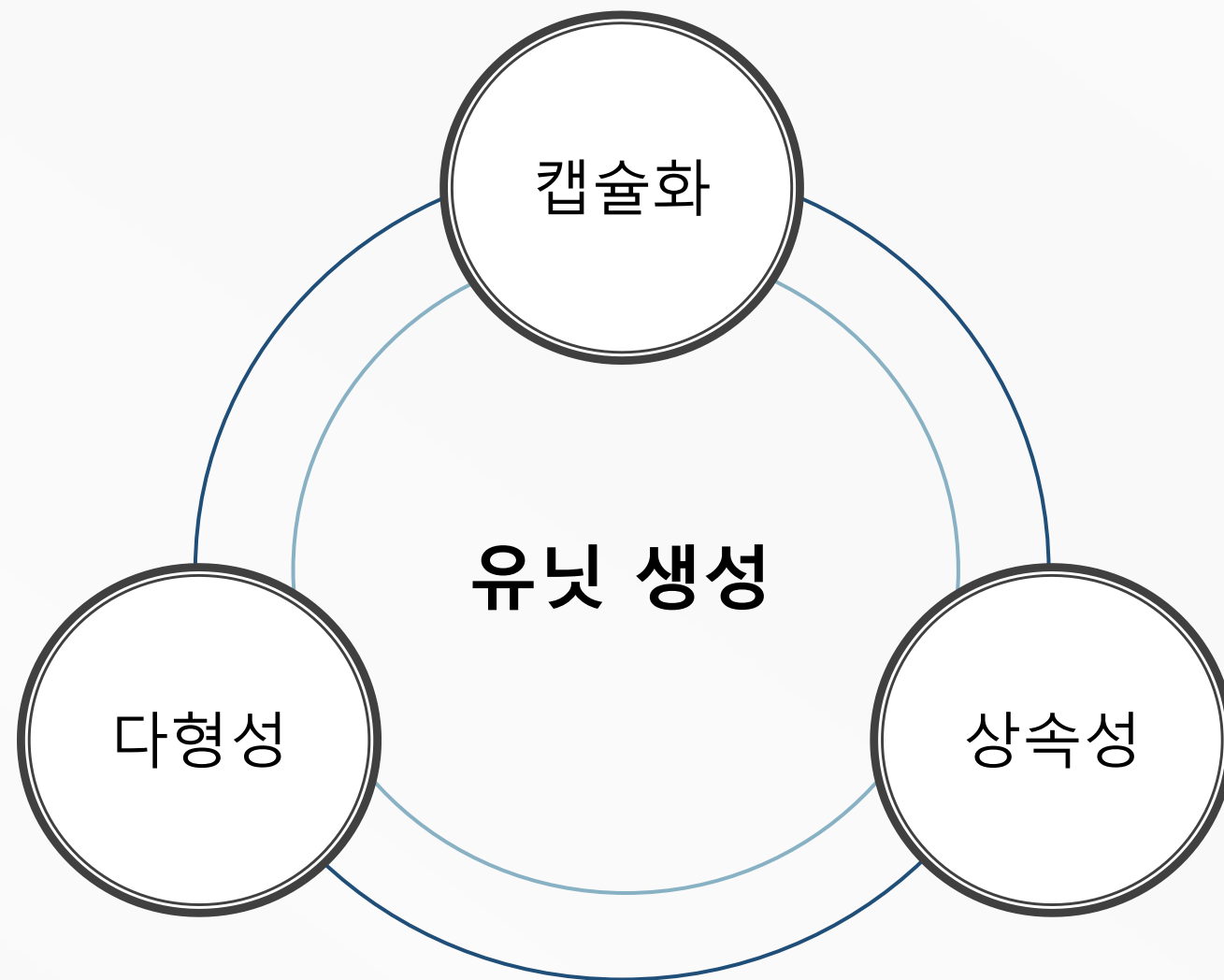
**객체지향을 이용한**

**STAR CRAFT**

**유닛 만들기**

이정훈

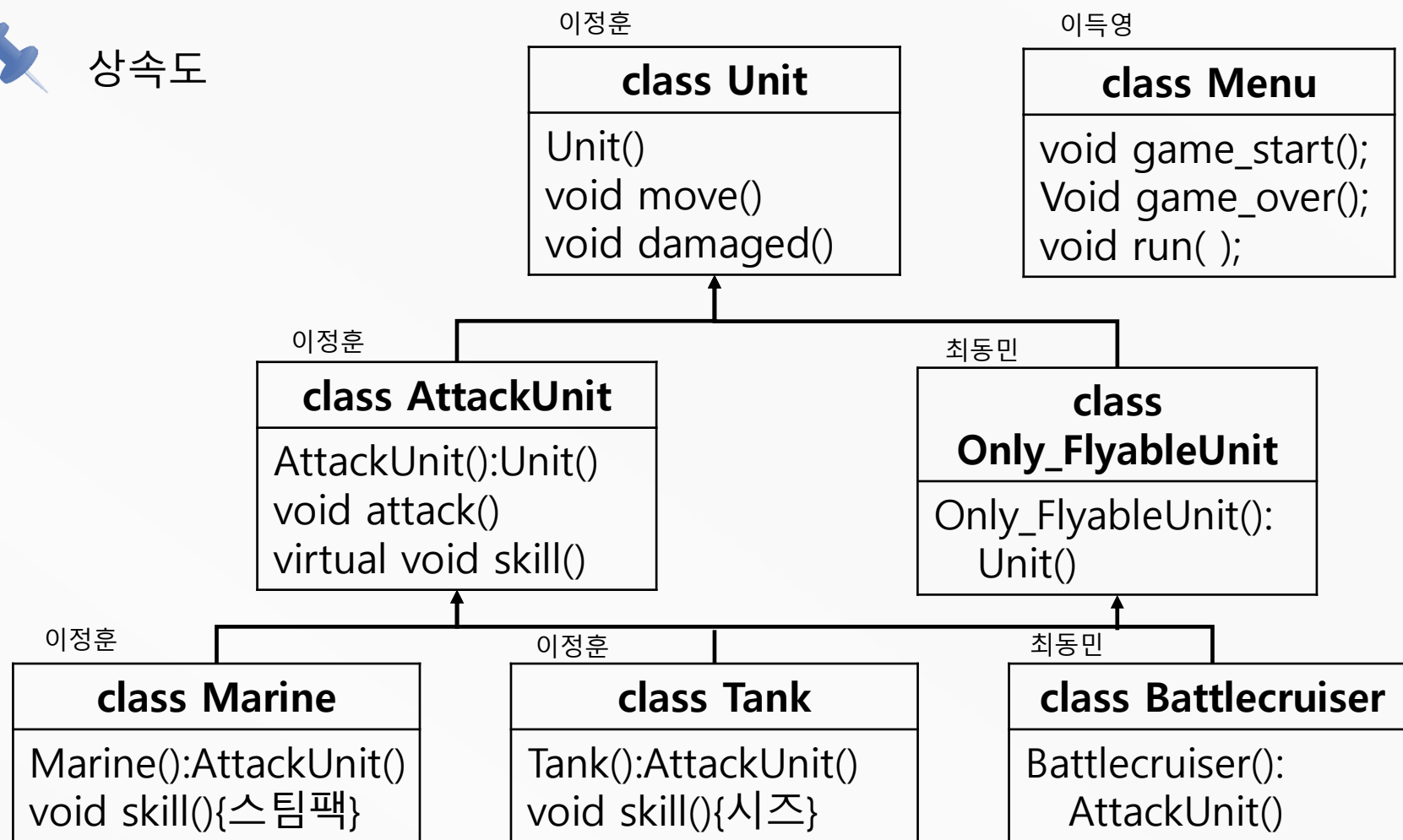
# 과제 목적



# 시스템 기능 구성도



상속도



# 시스템 기능 구성도



상속성

```
Marine m2("마린", 40, 5, 10);  
Tank t2("탱크", 100, 10, 30);  
m2.move(move_vector);  
t2.move(move_vector);  
m2.attack(move_vector);  
t2.attack(move_vector);
```

# 시스템 기능 구성도



## 캡슐화

```
class Unit {  
protected:  
    string name;  
    int speed;  
    int hp;  
public:  
    Unit() {  
        name = "marine";  
        hp = 40;  
        speed = 1;  
    }  
    Unit(string name, int hp, int speed);  
    int get_hp();  
    void move(string location);  
    void damaged(int damage);  
};
```

# 시스템 기능 구성도



## 다형성

```
class AttackUnit : public Unit {
protected:
    int damage;
public:
    void Attack_Unit_skill();
    virtual void skill();
    AttackUnit(string name, int hp, int speed, int damage);
    void attack(string location);
};
```

```
void Marine::skill() {
    if (hp > 10)
    {
        hp -= 10;
        damage *= 2;
        cout << name << " : 스팀팩을 사용합니다. ( HP 10 감소, 공격력 2배 [ 현재 체력 : " << hp << " ] ) << endl;
    }
    else
    {
        cout << name << " : 체력이 부족하여 스팀팩을 사용 할 수 없습니다." << endl;
    }
}
```

```
void Tank::skill() {
    if (siege_developed == false)
    {
        damage *= 2;
        speed = 0;
        cout << name << " : 시즈모드로 전환합니다. 공격력 3배 속도 0 [ 공격력 : " << damage << " ] << endl;
    }
    else
    {
        cout << name << " : 시즈모드를 해제합니다." << endl;
        damage /= 2;
    }
}
```

# 시스템 기능 구성도



## 업캐스팅

```
AttackUnit* p_attack_unit = new Marine("마린", 40, 5, 10);  
p_attack_unit->Attack_Unit_skill();  
  
AttackUnit* p_attack_unit1 = new Tank("탱크", 100, 10, 30);  
p_attack_unit1->Attack_Unit_skill();
```

## 다시 목적으로 돌아가서

### 캡슐화

- 데이터+알고리즘
- 데이터 은닉

### 다형성

- skill( )함수를 통해

### 상속성

- Unit부터  
Marine, Tank,  
Battelcruiser



감사합니다.