

Abstraction

2017-05-25

Content

- 1. Necessity
 - Model?
- 2. Example
- 3. Definition
- 4. Implementation
 - Polymorphism?

Necessity

- 왜 추상화를 알아야하는가?
 - 현실의 정보 중 인간이 관심있는 부분을 컴퓨터 내부로 이식하기 위해
 - 실세계는 복잡하기 때문에 이를 바로 컴퓨터 내부로 반영할 수는 없음
 - 추상화를 통해 실세계를 간결하고 명확히 “모델링”하게 됨

Model?

- 1. A standard or example for imitation or comparison
- 2. A representation, generally in miniature, to show the construction or appearance of something

Model?

- 1. 모방 혹은 비교를 위한 기준 또는 예시
- 2. 일반적으로 간소화된, 어떠한 것의 구조나 형태를 보이기 위한 표현

Modelling!

- 현실에 실재하는 복잡한 현상을 다수의 관점으로 분리함
- 각 관점에서 요구하는 필수적인 것에만 초점을 맞추어 생각함

Example



- 구직자 A씨가 B사에 지원하려 합니다.
- 이때 B사가 필요로 할 A씨의 정보는?

Example



- 구직자 A씨가 B사에 지원하려 합니다.
- 이때 B사가 필요로 할 A씨의 정보는?
- 신규직의 경우 :
 - 대학 졸업여부, 대졸 시 평균학점
 - 자격증, 시험성적
 - 공모전 등의 수상경력 등

Example



- 구직자 A씨가 B사에 지원하려 합니다.
- 이때 B사가 필요로 할 A씨의 정보는?
- 경력직의 경우 :
 - 이전 근무 기간, 직급
 - 업무 처리 역량을 증명가능한 그 어떠한 것

Example



- 이 과정에서 A씨의 성별, 외모, 음악 취향 등등...
- 굳이 필요하지 않은 정보들은 회사가 알 필요가 없음
- 이와 같이, 복잡한 실생활의 개체에서 사용자가 원하는 필수적인 정보만을 골라내는 것!

Definition

- 복잡한 자료, 모듈, 시스템 등으로부터 핵심적인 개념 또는 기능을 간추려 내는 행위
 - - 한국어 위키백과.
- 실세계의 복잡한 상황을 간결하고 명확하게 단순/일반/개념화
 - 구체적인 사항은 되도록 생략하고 핵심이 되는 원리만을 추구
 - - 정보통신기술용어해설

Implementation

- Class, Instance, Inheritance...
- Polymorphism!

```

int main()
{
    Unit* a = new GroundUnit();
    Unit* b = new AerialUnit();

    a->Move();
    b->Move();

    delete a;
    delete b;

    return 0;
}

```

```

class Unit
{
public:
    virtual void Move() = 0;
};

class AerialUnit : public Unit
{
    void AerialMove() { std::cout << "Flying" << std::endl; }

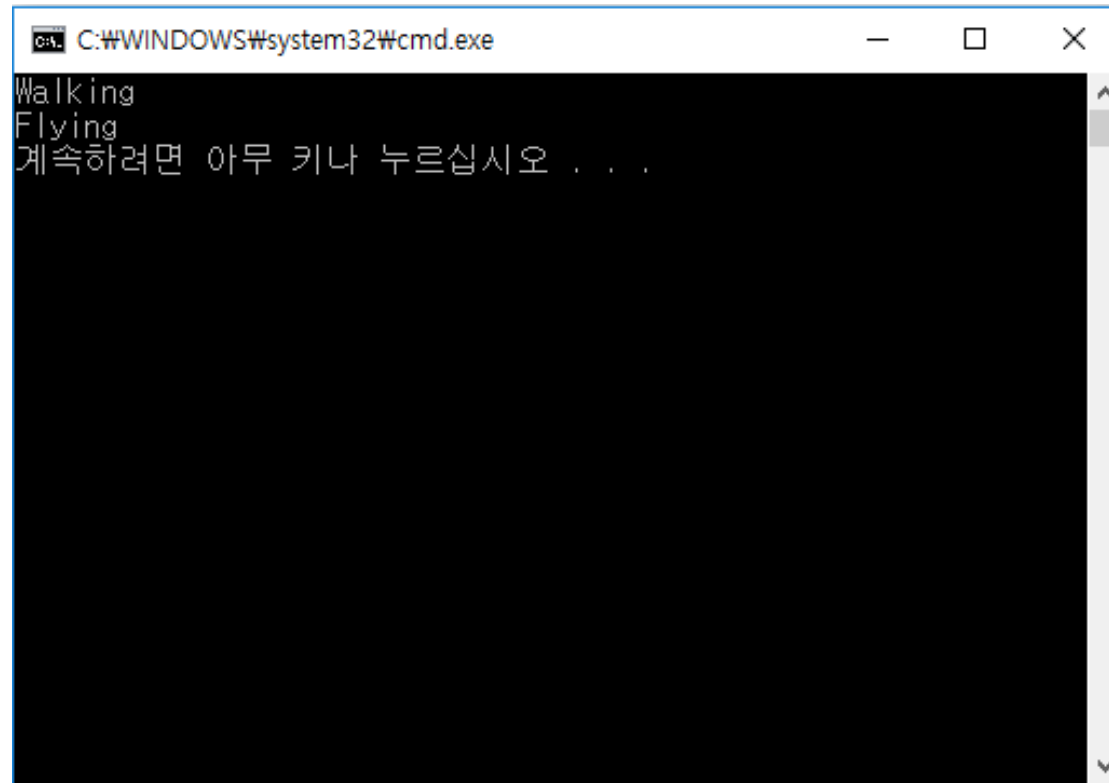
public:
    void Move() { AerialMove(); }
};

class GroundUnit : public Unit
{
    void GroundMove() { std::cout << "Walking" << std::endl; }

    void Move() { GroundMove(); }
};

```

Polymorphism?



```
C:\WINDOWS\system32\cmd.exe
Walking
Flying
계속하려면 아무 키나 누르십시오 . . .
```

Polymorphism?

Conclusion

- 추상화의 필요성
- 추상화의 대략적인 개념