

3장. 타입과 추상화

| 현상은 복잡하다. 법칙은 단순하다. 버릴 게 무엇인지 알아내라

중요한 것은 ?

추상화

🤔 왜 하는가 ?

🤔 어떻게 하는가 ?

객체지향 패러다임에서의 추상화

개념의 관점

분류

객체지향 설계

1 어떤 객체가 어떤 타입인가 ?

2 객체의 내부적인 표현은 외부로부터 철저하게 감춰진다.

3 타입은 계층을 갖는다.

🔄 일반화/특수화 관계 (슈퍼타입/서브타입)

정적 모델

🤔 왜 타입을 쓰는가 ?

타입 == 추상화

동적 모델과 정적 모델

동적 모델

정적 모델 (= 클래스)

중요한 것은 ?

연결, 즉 열차를 갈아타는 것입니다.

1. 꼭 알아야 하는 사실은 정확하게 표현
2. 몰라도 되는 정보는 무시함

→ 이해하기 쉽고 단순하며 목적에 부합하는 프로젝트

추상화

- 현실에서 출발하되 불필요한 부분을 도려내가면서 사물의 놀라운 본질을 드러나게 하는 과정
- 어떤 양상, 세부 사항, 구조를 좀 더 명확하게 이해하기 위해 특정 절차나 물체를 의도적으로 생략하거나 감춤으로써 복잡도를 극복하는 방법

🤔 왜 하는가 ?

- 불필요한 부분을 무시 → 현실에 존재하는 복잡성 극복
- 복잡한 현실을 단순화하기 위해 인간이 사용하는 인지 수단

🤔 어떻게 하는가 ?

1. 구체적인 사물들 간의 공통점은 취하고 차이점은 버리는 일반화를 통해 단순하게 만들기
2. 중요한 부분을 강조하기 위해 불필요한 세부 사항을 제거함으로써 단순하게 만들기

→ 모든 경우에 “이해하기 쉬운 수준으로 단순화” 할 것

객체지향 패러다임에서의 추상화

- **개념(Concept)** : 공통점을 기반으로 객체들을 묶기 위한 그릇
- **분류(Classification)** : 개념을 이용하여 객체를 여러 그룹으로 나누는 것
- **인스턴스(Instance)** : 객체에 어떤 개념을 적용하여 개념 그룹의 일원이 되는 객체
 - 객체는 특정한 개념을 적용할 수 있는 구체적인 사물
 - 인스턴스는 개념이 적용된 객체

개념의 관점

1. **심볼** : 개념을 가리키는 간략한 이름이나 명칭 트럼프
2. **내연** : 개념의 완전한 정의 몸이 납작하고 ~
3. **외연** : 개념에 속하는 모든 객체의 집합 정원사, 병사, ..., 하트 여왕

타입

컴공들이 개념을 좀 더 간지나보이게 하기 위해 만든 용어

ex) 데이터 타입 : 메모리는 0과 1만 존재 → 타입이라는 질서가 없음

- 질서를 만들기 위해 메모리 안의 데이터에 의미를 부여한 것

타입 시스템

- 메모리 안의 데이터로 야기되는 혼란을 방지하기 위해 만들어짐
- 데이터가 잘못 사용되는 것을 방지

결론

1. 타입 = 데이터가 어떻게 사용되느냐
2. 타입에 속한 데이터를 메모리에 어떻게 표현하는지는 비밀이다.

분류

객체에 특정한 개념을 적용하는 작업

- 객체지향의 가장 중요한 개념이기도 함. **Class가 객체 지향 세계의 유명 인사니까 !**
- 어떤 객체를 어떤 개념으로 분류할지가 객체지향의 품질을 결정

객체지향 설계

데이터를 먼저 결정하고 객체의 책임을 결정하는 방법(Data-Driven Design)은 유연하지 못하다 !

1 어떤 객체가 어떤 타입인가 ?

- 객체가 수행하는 행동이 결정
- 어떤 객체들이 동일한 행동을 수행할 수 있다면 동일한 타입으로 분류 가능 (다형성)

2 객체의 내부적인 표현은 외부로부터 철저하게 감춰진다.

- 효과적으로 수행할 수만 있다면 내부의 상태를 어떻게 표현해도 무방 (캡슐화)

3 타입은 계층을 갖는다.

트럼프 인간 타입의 외연은 트럼프 타입의 외연의 부분 집합으로 표현할 수 있다.

☞ 일반화/특수화 관계 (슈퍼타입/서브타입)

객체의 상태를 표현하는 데이터가 아니라 행동에 따라 관계가 결정된다.

- 특수한 타입은 일반적인 타입이 할 수 있는 모든 행동을 동일하게 수행해야 함
- 타입의 내연을 의미하는 행동 수와 외연을 의미하는 집합 크기는 서로 반대다.
 - 일반적 타입은 특수 타입보다 더 적은 수로 행동하지만 더 큰 크기의 외연 집합을 갖는다.

정적 모델

왜 타입을 쓰는가 ?

인간의 인지 능력으로 시간에 따라 동적으로 변하는 객체의 복잡성을 극복하기 어렵다.

- 타입을 통해 정적인 모습으로 상태를 다룰 수 있다.
- 복잡성을 제거하여 시간에 독립적인 객체를 생각할 수 있다.

타입 == 추상화

- 타입을 이용해 객체의 동적인 특성을 추상화한다.
- 타입은 객체의 상태 변경이라는 복잡성을 단순화한다.

동적 모델과 정적 모델

동적 모델

- 객체가 특정 시점에 구체적으로 어떤 상태를 가지는가 ?
- 객체의 스냅샷

정적 모델 (= 클래스)

- 객체가 가질 수 있는 모든 상태와 모든 행동을 시간에 독립적으로 표현
- 타입 모델
 - 타입을 구현하는 가장 보편적인 방법은 클래스 !
 - 💡 타입 ≠ 클래스

