

3. 타입과 추상화

지하철 노선도 추상화 → 헨리백 (역사이 연관관계에 집중)

= 꼭 알아야하는 정보만!

→ 자형정보 버리고 단순해짐.

= 이해하기 쉽고 단순하게

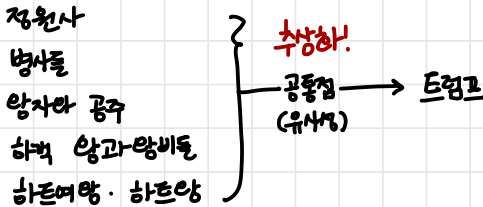
1) 추상화를 통한 복잡성 극복

이해하기 쉽고 예측 가능한 수준으로 현실을 분해하고 단순화

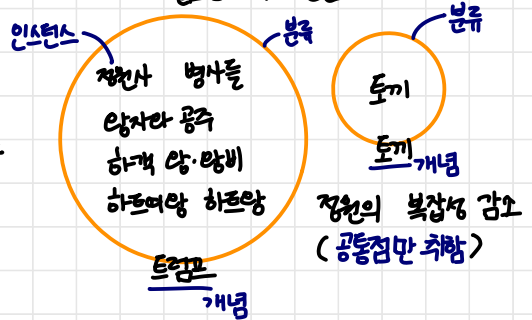
추상화: 복잡한 현실 → 단순화 (의도적 생략, 감축)

2) 객체지향과 추상화

2-1. 모두 트럼핀본



2-2. 그룹으로 나뉘어 단순화 하기



2-3. 개념

• 개념 (concept) = 공통점을 기반으로 객체들을 묶기 위한 그릇.

• 분류 (classification) = 개념을 이용해서 여러 그룹으로 분류 가능

트럼프 / 토끼 개념 나누고 분류함.

• 인스턴스 (instance) = 객체가 개념에 분류되면, 객체는 그 개념의 인스턴스이다.

추상화: 객체 → 몇가지 개념의 인스턴스

2-4. 개념의 세가지 관점

- 심볼 (Symbol): 개념을 가리키는 간략한 이름이나 명칭
- 내면 (intension): 개념의 완전한 정의.
내면으로 객체가 개념에 속하는지 여부확인
- 외면 (extension): 개념에 속하는 모든 객체의 집합

외면 —

정확한 명사들
양자라 공주
하객 양. 양비
하드웨어 하프양

트렌드

심볼

2-5. 객체 분류하기 위한 틀

개념 분류 → 객체지향 품질 결정
: 유지 보수 good

내면 = 몸 압착 /

두 손발이 네모난 몸모서리

2-6. 분류는 추상화를 위한 도구이다.

분류: 복잡 → 단순

3) 타입

3-1. 타입은 개념이다.

개념 (concept) = 타입 (type) → 기계적 의미 하강!
→ 공통점 기반으로 객체 묶기.

3-2. 데이터 타입

- 타입시스템 (type System) : 0.1 출산 방지
→ 데이터가 어떻게 사용되느냐. → 자료구조: 데이터 타입 + 연산자
: 연산자 - 데이터
→ 타입에 속한 데이터를 메모리에 어떻게 표현하는지 외부로부터 철저히 감춤.
: 저장 방식 몰라도 Integer 형 데이터 가능. → 우리는 Integer 저장 구조 모름

3-3. 객체와 타입

객체 ≠ 데이터
= 객체의 행동이 중요!

→ 데이터 타입에 객체 이야기 그대로!

→ 어떤 객체가 어떤 타입에 속하는지 결정하는건 객체가 수행하는 행동이다.

→ 객체의 내부적인 표현은 외부로부터 철저히 감춰진다.

3-4. 행동이 우연이다.

객체 타입 결정 — 객체의 행동 (only)

동일한 행동 = 동일한 책임
= 동일한 메시지 수신

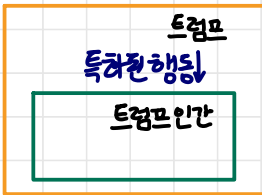
같은 타입인 객체 = 행동이 동일하면 서로 다른 데이터 가질 수 있다.
→ 데이터 달라도 동일 메시지 수신 가능

캡슐화 : 외부에 행동만 제공! ← 먼저생각 (행동으로 객체 분류)
데이터는 감춘다.

데이터 주도 설계 (Data Driven Design) $\xrightarrow{\text{보안}}$ 책임 주도 설계 (Responsibility Driven Design)

4) 타입의 계층

4-1. 드림프 계층



드림프 인간 타입 객체는
드림프 객체에도 속해야 한다.

일반화 / 특수화 (Generalization / Specialization)

4-2. 일반화 / 특수화

일반적 타입 : 더 적은수 행동

특수한 타입 : 더 많은 행동

