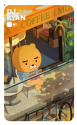

⭐Kakao Cloud(Day 5)⭐

Class란 무엇인가?

1. **객체 모델링 수단**
2. **instance를 만들어 내는 도구**
3. **ADT**

> 💡 변수와 함수 등 여러개의 데이터 타입을 이용해서 새로운 하나의 데이터 타입을 만드는데 이를 ADT(Abstract Data Type)라고 한다.

⭐구조적 프로그래밍(절차적)→ C언어⭐

1. 프로그램을 기능적으로 세분화 시킨다.
2. 기능을 모듈화 시킨다 → function(함수)
   - ⇒ 프로그램을 쉽고 빠르게 구현(설계가 비교적 간단)
   - ⇒ But 유지보수가 힘들다.
   - 수정하게 되면 연결되어있는 다른 프로그램에 문제가 생길 수 있다.

⭐객체지향 프로그래밍⭐

객체지향 개념을 가장 잘 대변 하는 언어는 **JAVA**

- ⇒ 유지보수성이 높다. 프로그램을 기능적으로 세분화하지 않는다.
- ⇒ 현실세계에 내가 해결해야 하는 문제를 program적으로 표현(modeling)
- ⇒ 문제를 구성하는 구성요소를 파악 → 그 구성요소간에 어떤 data가 오고가는지 파악 →

  프로그램으로 묘사

⭐Javascript Class의 특징⭐

생성자 함수와 유사 → class는 함수이면서 객체→ instance를 생성 가능

- class는 반드시 “new” keyword와 함께 사용
- class간의 상속을 지원 → extends, super
- class역시 hoisting이 발생 (let,const처럼 hoisting이 된다)
- class 내부코드는 String mode로 동작한다.
- class 안에는 **constructor, prototype method, static method**
  - → 이 property의 property attribute 값은 모두 false이다.

Class는 0개 이상의 method로 구성

- “ES6” 축약표현으로 된 method(non-constructor)
- constructor method(생성자)
  - 반드시 1개만 가능, instance를 만들고 초기화 해주는 역할
- prototype method
- static method

⭐Inheritance(상속)⭐

▼ Super class

super class (슈퍼 클래스)

parent class (부모 클래스)

upper class (상위 클래스)

base class (기반 클래스)

▼ Sub class

sub class (서브 클래스)

child class (자식 클래스)

derived class (파생 클래스)

Is-A relationship

→ Sub class is a Super class

- class 간 상속은 가능!
- 생성자 함수간에 상속은 안된다
- class와 생성자 함수는 상속관계를 가질 수 없다.
- 동적상속은 허용한다.
- extends 키워드 뒤에 값으로 평가될 수 있는 식이 올 수 있다.

```
class Base {
  constructor(name) {
    this.name = name;
  }

  sayhello() {
    return "안녕하세요!";
  }
}

class Derived extends Base {
  sayhello() {
    return super.sayhello() + this.name;
  }
}
const base = new Derived("신재민");
console.log(base.sayhello());
```

Array(배열)

가장 대표적으로 많이 사용 되는 자료구조 ⇒ 활용빈도가 아주 높다!

▼ **array literal → [ ]**

- const arr = [1,2,3] // array literal
- const arr = new Array();

▼ **객체**

- property key
- property value
- 순서가 없다
- length가 없다.

▼ **배열**

- index 요소
- 순서가 있어요
- length가 있다.

Data Struture 관점에서 Array

1. 연속적인 저장공간
2. 같은 data type을 가져요(각 공간의 크기가 같아요) ⇒ dense array(밀집 배열)
    - index를 이용해 직접적으로 값을 빠르게 access한다 !
    - But, 삽입과 삭제처리는 비용이 많이 든다.
        - 연속적인 공간이기 때문에 삽입,삭제를 했을 때
        - 뒤에있는 데이터가 당겨지거나 밀리기 때문

그런데 JavaScript 배열은 Sparse array(최소 배열)이기 때문에 일반적인 배열의 특징과는 반대되는 특징을 가지고있다.