

Ch4. 객체



객체(Objects)

- 게임의 상태를 지속적으로 파악하기 위한 **Game** 클래스가 필요하다고 가정
- **Game** 클래스의 인스턴스를 많이 생성한다면 문제가 됨
- 왜냐하면 각 인스턴스는 상호 동기화되기 어려운 자신만의 상태를 가질 수 있음
- 또한, 너무 많은 인스턴스가 생성되면 자원이 낭비됨
- 위의 상황에서 사용되는 방법이 싱글톤(**Singleton**)
- 코틀린에서는 **object** 키워드를 사용하면 싱글톤 객체를 만들 수 있음
- **object** 키워드를 사용하는 방법은 객체 선언, 객체 표현식, 동반 객체 세 가지가 있음

중첩 클래스(Nested Class)

- 다른 클래스 내부에 중첩된 클래스가 존재
- 특정 클래스 내부에서만 사용될 클래스를 정의하기 위해 사용
- 중첩 클래스의 인스턴스는 이 클래스를 포함하는 외곽 클래스의 인스턴스가 생성되어야 사용 가능
- 외곽 클래스에서는 중첩된 클래스의 속성과 함수를 사용할 수 있음

데이터 클래스(Data Class)

- 데이터를 저장하기 위해 설계된 클래스
- 강력한 데이터 처리 기능을 갖고 있음
- 선언 : `data class` 클래스 이름
- `toString()`, `equals()`, `copy()` 등의 메소드를 포함

감사합니다^^

강사 목진혁

jhmocu@gmail.com