객체지향의 이해 (1) - 연습문제

🐾 동물원 사파리 대탐험!

📝 문제 설명

당신은 동물원의 사육사입니다.

동물들의 공통적인 특징과 각각의 독특한 행동을 자바 클래스로 표현하려고 합니다.

다음 조건에 따라 클래스를 작성하세요.

💡 요구사항

1. `Animal` 클래스를 만든다.
    - 메서드 `move()`와 `makeSound()`를 포함한다.
    - 이 클래스의 멤버변수와 생성자는 없다.
2. `Lion`, `Elephant`, `Penguin` 클래스를 `Animal` 클래스를 상속하여 만든다.
    - 각 클래스에서 `move()`와 `makeSound()`를 **오버라이딩**한다.
    - 예시 동작:
        - **사자**: 네 발로 달려가며, 포효함
        - **코끼리**: 천천히 걸으며, 트럼펫처럼 울음
        - **펭귄**: 미끄러지듯 이동하며, 삐약거림
3. `Animal` 타입의 배열을 만들어 여러 동물을 저장한 후,
   `for`문을 통해 모든 동물의 `move()`와 `makeSound()`를 호출해보자.
4. `Animal`, `Lion`, `Elephant`, `Penguin` 클래스는 default 클래스 입니다.

🧪 출력 예시

```
사자가 네 발로 달려갑니다.
사자가 포효합니다!
코끼리가 천천히 걷습니다.
코끼리가 뿌웁~ 하고 울어요!
펭귄이 미끄러지듯 이동합니다.
펭귄이 삐약삐약 울어요!
```

☑️ 학습 목표

- 클래스 **상속**의 구조 이해
- 메서드 **오버라이딩** 구현
- **다형성(polymorphism)** 활용
- **업캐스팅**과 배열을 통한 객체 관리

🌟 도전 과제 (선택)

- 펭귄 클래스에 `swim()` 메서드를 추가해보고,
  `instanceof` 와 **다운캐스팅**을 이용해 수영 기능도 호출해보세요!

🛒 연습문제 2 – 쇼핑몰 상품 클래스 만들기

📝 문제 설명

당신은 온라인 쇼핑몰의 개발자입니다.

쇼핑몰에 등록된 상품은 모두 `Product` 클래스를 기반으로 만들며,
전자제품, 의류, 식료품 등 다양한 상품들이 존재합니다.

각 상품은 공통적인 동작을 가지되, 고유한 설명을 가지고 있습니다.

클래스를 활용해 이 구조를 자바 코드로 구현해보세요.

💡 요구사항

1. `Product` 클래스를 만든다.
    - `showInfo()` 메서드를 포함하며, "상품 정보 출력"이라는 기본 메시지를 출력한다.
    - 이 클래스의 멤버변수와 생성자는 없다.
2. `Electronics`, `Clothing`, `Food` 클래스를 `Product` 클래스를 상속하여 만든다.
    - 각 클래스는 `showInfo()`를 **오버라이딩**한다.
    - 예:
        - 전자제품: "전자제품입니다. 최신 기기!"
        - 의류: "옷입니다. 계절별 신상품!"
        - 식료품: "식품입니다. 유통기한을 확인하세요!"
3. `Product` 타입의 배열을 만들어, 여러 상품 객체를 저장한 뒤,
   `for`문을 통해 모든 상품의 `showInfo()`를 호출한다.
4. Product, Eletronics, Clothing, Food 클래스는 public 클래스 입니다.

🧨 출력 예시

```
전자제품입니다. 최신 기기!
옷입니다. 계절별 신상품!
식품입니다. 유통기한을 확인하세요!
```

☑️ 학습 목표

- **상속**을 통한 구조 설계 연습
- **메서드 오버라이딩**으로 동작 재정의

- **다형성(polymorphism)** 적용
- **업캐스팅**, 객체 배열 활용

✨ 도전 과제 (선택)

- `Food` 클래스에 `checkExpiration()` 메서드를 추가하고,
  `instanceof`와 **다운캐스팅**을 통해 호출해보세요.

📜 연습문제 3 – 음식 주문 시스템 (심화 버전)

📝 문제 설명

당신은 음식 주문 앱의 개발자로, 다양한 음식 메뉴를 클래스로 설계하고 있습니다.
모든 음식은 공통된 속성을 가지지만, 각자 고유한 설명과 옵션 추가 기능이 존재합니다.
객체지향 개념을 바탕으로 클래스를 설계하고, 각 음식 고유의 동작을 구현해보세요.

💡 요구사항

1. `MenuItem` 클래스를 만든다.
   - 멤버 변수: `name`, `price`
   - 생성자, getter, setter 포함
   - 메서드:
     - `toString()` → "이름: 메뉴 설명"
     - `addOption()` → "옵션 추가 기능은 각 음식에 따라 다릅니다." (기본 메시지)
2. `Burger`, `Pizza`, `Salad` 클래스를 `MenuItem` 클래스를 상속하여 만든다.
   - 각각 `toString()`과 `addOption()`을 **오버라이딩**한다.
     - `Burger`: "치즈 추가"
     - `Pizza`: "엣지 변경: 치즈 크러스트"
     - `Salad`: "드레싱 선택: 발사믹"
3. `MenuItem` 타입의 배열을 만들어, 다양한 음식 객체를 저장하고,
   `for`문을 통해 설명(`toString`)을 출력하고, 총 가격을 계산한다.
4. 각 음식 객체를 `instanceof`로 판별하고 **다운캐스팅**하여 `addOption()` 메서드를 호출한다.

🧪 출력 예시

```
치즈버거: 패티와 빵이 조화를 이루는 메뉴
페퍼로니 피자: 치즈 듬뿍, 모두의 인기 메뉴
그린 샐러드: 신선한 채소로 만든 건강식
총 가격: 19000원
옵션 적용:
    치즈버거 → 치즈 추가
    페퍼로니 피자 → 엣지 변경: 치즈 크러스트
    그린 샐러드 → 드레싱 선택: 발사믹
```

☑ 학습 목표

- 상속 및 메서드 오버라이딩 심화
- 객체 배열을 통한 다형성 구현
- 다운캐스팅과 고유 기능 호출
- 실생활 개념의 객체지향적 설계 연습