1. 문제 정의

프린터는 종이와 잉크(혹은 토너)라는 소모품이 필요하며, 이러한 소모품이 부족하면 인쇄를 할 수 없습니다. 두 가지 종류의 프린터가 있으며 각각은 다른 소모품을 사용합니다. 사용자가 인쇄를 요청할 때마다 종이와 잉크/토너의 부족 여부를 체크하여 인쇄 작업을 처리해야 합니다. 이 과정에서 사용자가 계속해서 인쇄 작업을 시도할 수 있도록 하되, 각 소모품이 부족한 경우 적절한 메시지를 출력해야 합니다.

2. 문제 해결 방법

- 두 가지 종류의 프린터(InkJetPrinter, LaserPrinter)를 각각 클래스로 구현한다.
- 각 프린터는 인쇄 가능한 페이지 수와 남은 잉크/토너 또는 종이 수를 추적한다.
- InkJetPrinter는 인쇄 시 종이와 잉크의 양을 모두 체크하며, LaserPrinter는 종이와 토너 양을 체크한다.
- 각 프린터 클래스는 공통된 부모 클래스를 상속받아 기본적인 속성(모델명, 제조사, 종이 수, 인쇄 횟수 등)을 공유하고, print()라는 가상 함수를 통해 각 프린터의 인쇄 로직을 다르게 구현한다.
- 사용자 입력에 따라 각 프린터에서 인쇄 작업을 처리하고, 인쇄 후 잉크나 토너, 종이 수를 출력하여 사용자가 상태를 확인할 수 있도록 한다.
- virtual과 protected를 적절히 사용하여 상속을 활용하고, 다형성을 적용하여 코드를 확장 가능하고 유지보수 용이하게 만든다.

3. 아이디어 평가

<장점>
- **상속과 다형성 사용**: 부모 클래스 Printer를 정의하고, 이를 상속받은 InkJetPrinter와 LaserPrinter클래스에서 각각의 세부 동작을 다르게 구현하여 코드의 중복을 줄였다. 부모 클래스에서 공통된 속성을 정의하고, 각 프린터에 맞는 동작만 오버라이드하는 방식은 객체 지향적 설계에서 좋은 예시이다.
- **유연성**: 추후에 프린터 종류가 추가되더라도 Printer클래스를 상속받고 print()메서드를 오버라이드하는 방식으로 쉽게 확장할 수 있다.
- **메모리 관리**: new와 delete를 사용해 동적 메모리 할당과 해제를 통해 객체를 효율적으로 관리한다.

<단점>
- InkJetPrinter와 LaserPrinter클래스가 protected접근 지정자를 사용하고 있기 때문에, 상속을 통해서만 접근 가능하다. 이렇게 되면 외부에서 직접 model, manufacturer, availableCount등의 변수에 접근할 수 없게 되어 코드 유지보수에 있어 불편할 수 있다.
- 각 프린터 클래스에서 print()메서드와 별도로 상태를 출력하는 함수(printInkJet, printLaser)가 중복되어 있다. 이는 하나의 printStatus()같은 함수로 통합할 수 있었을 것이다.

4. 문제를 해결한 키 아이디어 또는 알고리즘 설명

소모품 체크: 각 프린터의 print()함수에서는 인쇄 가능한지 여부를 체크합니다.

- **잉크젯 프린터**: availableCount(종이 수)와 availableInk(잉크 수)를 비교하여, 둘 중 하나라도 부족하면 인쇄를 못 하도록 처리합니다.
- **레이저 프린터**: availableCount(종이 수)와 availableToner(토너 수)를 비교하여, 토너는 한 페이지당 0.5만큼 소모되는 것으로 계산하여 부족 여부를 판단합니다.