

1) 문제 정의

상속 구조 클래스를 사용한다.

모든 프린터는 모델명(model), 제조사(manufacturer), 인쇄 매수(printedCount), 인쇄 종이 잔량(availableCount),을 나타내는 정보와 print(int pages) 멤버 함수를 가지며, print 가 호출할 때마다 pages 의 용지를 사용해야 한다.

잉크젯 프린터는 잉크 잔량(availableInk) 정보와 printInkJet(int pages_멤버 함수를 추가적으로 가지며,

레이저 프린터는 토너 잔량(availableToner) 정보와 PrintLaser(int pages) 멤버 함수를 추가적으로 가진다.

1 개의 부모 클래스 프린터와, 그 프린터를 상속받는 2 가지 서로 다른 변수와 함수를 가지는 클래스를 만들어서 객체로 만든 후 사용한다.

2) 문제 해결 방법

먼저 문제에 클래스에 들어가야 하는 변수와 함수를 모두 정해 주어서 그것들만 사용해서 만들도록 한다.

메인 함수는 객체를 1 번씩 만들고 반복문을 사용해 돌리면 된다고 생각한다.

클래스별 생성자 함수는 들어가야 하는 매개변수가 있기 때문에 필요하다고 생각하지만 소멸자 함수는 따로 쓸 일이 없다고 생각해서 기본 소멸자 함수로 충분할 것으로 생각되어 따로 구현하지는 않을 듯하다.

변수 관련해서는 모두 protected 를 사용할 듯하다.

3) 아이디어 평가

위 문제 해결 방법에 적합한 방식으로 처리했는데 print 함수와 잉크프린터 클래스, 레이저프린터 클래스에 있는 함수 간에 역할 배분과 예외 처리에 힘이 많이 들었다.

처음에는 print 함수에서 프린터 클래스의 printedCount 와 availableCount 를 조작하고, 자손 클래스의 함수들에서는 이를 비교하며 예외 처리를 하려고 했지만 예외처리에 문제가 계속 생겨서 위 두 변수의 조작을 자손 클래스의 함수로 위치를 옮겼다.

또 문제에서 사용한 것 중 printedCount 변수는 작성하고 프린트 시에 수가 올라가도록 설정되도록 만들었지만, 문제상 딱히 사용하는 곳이 없어 지워도 되지 않을까 싶었다.

main 함수에서 반복문을 돌릴 때, 또 입력을 받을 때 생각보다 많은 예외 처리가 있어야 프로그램이 정상적으로 작동하여 이런 요소들을 초반에 전혀 생각하지 못했던 것이 아쉬웠다.

3) 알고리즘 설명

클래스 선언은 문제에 나와있는 방식으로 그대로 작성하였고, 클래스마다 생성자 함수와 상속 값 설정만 해 주었다.

부모 클래스인 Printer 클래스는 Printer 생성자 함수에 model, manufacturer availableCount 를 매개변수로 받았고, 이를 초기화 해주도록 생성자 함수를 작성하였다.

Print 함수는 pages 를 인수로 받고 availableCount 와 pages 를 비교하여 프린트 성공 여부를 출력하도록 만들었다.

매개변수는 자식 클래스에서 넘겨받는 식으로 넣어주게 될 것이다.

InkPrinter 클래스와 LaserPrinter 클래스는 변수와 함수, 생성자 함수의 이름만 다르고 알고리즘 자체는 완전히 동일하게 작성하였다.

InkPrinter 기준으로 설명하면,

먼저 생성자 함수는 model, manufacturer, availableCount, availableInk 를 매개변수로 받고, Printer 클래스에 model, manufacturer, availableCount 를 넘겨주면서 상속받는다. 위 프린터 클래스의 생성자 함수와 맞춰주었다. 기능은 availableInk 를 초기화하고, 모델, 사명, 남은 종이, 남은 잉크 정보를 출력해준다.

printInkJet 함수를 살펴보면 pages 를 인수로 받고, 페이지가 0 보다 작거나 같을 때는 앞에 말했던 출력문을 출력해주고, availableCount 가 pages 보다 크거나 같으면 availableCount 에서 pages 를 빼 주고, printedCount 를 pages 만큼 더해준 후 availableInk 를 pages 만큼 빼 준다. 그 후 최종적으로 앞의 출력문을 출력해 준다.

LaserPrinter 클래스는 거의 모든 것이 InkPrinter 와 같지만 출력문의 "잉크"라는 글씨가 "토너"로 바뀌었다는 것, availableToner 를 pages 만큼 빼 주는 것 대신 1 만큼 빼 준다. 이는 문제에서 잉크는 pages 만큼 빠지지만, 레이저는 1 씩 빠져서 이렇게 구성하였다.

두 클래스는 모두 public 으로 프린터 클래스를 상속받았다.

Main 함수에서는

기본 출력문을 작성해 주고, inkPrinter 와 laserPrinter 객체를 생성한다. 2 가지 수 mode 와 pages 를 입력 받는다. 페이지가 0 보다 크다면 mode 에 따라 잉크 모드라면 잉크프린터 객체의 print 함수와 잉크의 printInkJet 함수를 pages 매개변수를 넣어 실행해 준다. 그리고 레이저프린터 객체의 printLaser 함수는 0 을 넣어주어 출력문을 출력해 준다. 레이저 모드라면 반대로 해주면 된다.

그 후 프린터를 계속 작동시킬 것인지에 관한 입력을 받아준다.

객체를 선언한 이후부터 마지막까지 while true 문을 돌려주면서 반복하다가 프린터 작동에

관한 입력에 n 을 입력 받으면 프로그램을 종료시켜준다.