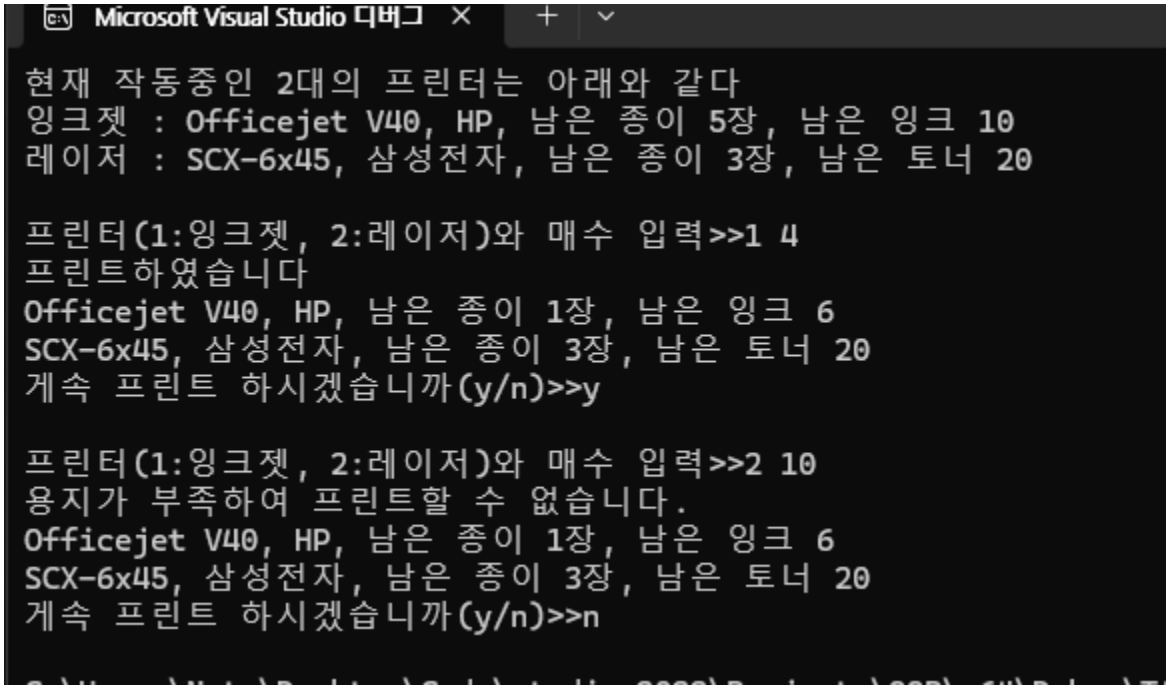
객체지향프로그래밍[02]
202304035 김유진
202104354 지현준

#실행 결과



#문제 정의

model(모델명), manufacturer(제조사)와 printedCount(인쇄 매수), availableCount(인쇄 종이 잔량)을 멤버 변수로 가지고, int 타입의 pages를 인자로 받는 print 멤버 함수를 가지는 Printer 클래스를 정의한다. Printer 클래스를 상속 받고, availableInk(잉크 잔량)를 멤버 변수로 가지며 int 타입의 pages를 인자로 받는 printInkJet 멤버함수를 가지는 InkJetPrinter 클래스를 정의한다. 또한 Printer 클래스를 상속 받고, availableToner(토너 잔량)를 멤버 변수로 가지며 int 타입의 pages를 인자로 받는 printLaser 멤버함수를 가지는 LaserPrinter 클래스를 정의한다.

#문제 해결 방법

Printer 클래스 : string 타입의 model(모델명), manufacturer(제조사)와 int 타입의 printedCount(인쇄 매수), availableCount(인쇄 종이 잔량)을 멤버 변수로 가지도록 한다. int 타입의 pages를 인자로 받는 print 멤버 함수를 가진다. 이때, model, manufacturer, availableCount 멤버변수는 외부에서 접근을 막아 무단으로 정보 수정을 하지 못하도록 protected로 선언한다. private이 아닌 protected로 선언하는 것은 상속 클래스에서 Printer의 멤버변수(model, manufacturer, availableCount)를 상속받아 사용하기 때문이다. 따라서 상속 클래스의 생성자를 호출할 때, 접근할 수 있어야 하므로 protected로 선언한다. (private의 경우, 상속 클래스에서 기본 클래스의 멤버 변수 접근 불가) print 멤버 함수와 생성자, 소멸자, printedCount 멤버 변수는 외부에서 접근할 수 있어야 하므로 public 으로 선언한다. 또한 availableCount를 protected로 선언했기 때문에 main 함수에서 프린터에 대한 정보를 출력할 때 접근할 수 있도록

public으로 int 타입의 availableCount을 리턴하는 getavailableCount라는 멤버 함수를 정의한다. print 멤버 함수는 "프린트하였습니다"라는 문구를 출력하고 pages장의 종이를 사용했으므로 인쇄 종이 잔량(availableCount)에서 pages를 뺀다.

InkJetPrinter 클래스 : Printer 클래스를 **public으로** 상속 받고, int 타입의 availableInk(잉크 잔량)를 멤버 변수로 가지며 int 타입의 pages를 인자로 받는 printInkJet 멤버함수를 가지는 를 정의한다. 생성자에서는 this로 this 포인터를 사용해 멤버 변수를 초기화했다. main 함수에서 InkJetPrinter에 대한 정보를 반복적으로 출력하고 멤버 변수들을 protected로 선언했기 때문에 InkJetPrinter의 정보를 출력하는 showIJPinfo라는 멤버 변수를 public으로 선언해 정의했다. printInkjet 멤버함수에서는 Printer 클래스를 상속 받았기 때문에 print를 호출해 "프린트하였습니다"를 출력하고 인쇄 종이 잔량(availableCount)을 처리한다. 또한 추가적으로 pages만큼의 잉크를 사용했기 때문에 availableInk에서 pages를 빼 남은 잉크를 처리하는 코드를 작성했다.

LaserPrinter 클래스 : 또한 Printer 클래스를 **public으로** 상속 받고, int 타입의 availableToner(토너 잔량)를 멤버 변수로 가지며 int 타입의 pages를 인자로 받는 printLaser 멤버함수를 가지는 를 정의한다. 성자에서는 this로 this 포인터를 사용해 멤버 변수를 초기화했다. LaserPrinter에 대한 정보를 반복적으로 출력하고 멤버 변수들을 protected로 선언했기 때문에 LaserPrinter의 정보를 출력하는 showLPinfo라는 멤버 변수를 public으로 선언해 정의했다. printLaser 멤버함수에서는 Printer 클래스를 상속 받았기 때문에 print를 호출해 "프린트하였습니다"를 출력하고 인쇄 종이 잔량(availableCount)을 처리한다. 또한 추가적으로 pages의 반만큼의 잉크를 사용했기 때문에 availableToner에서 pages를 빼 남은 토너양을 처리하는 코드를 작성했다.

main 함수 : InkJetPrinter 객체인 ijp는 "Office V40" 모델의 HP 잉크젯 프린터로, 용지 5장과 잉크 10을 초기 값으로, LaserPrinter 객체인 lp는 "SCX-6x45" 모델의 삼성전자 레이저 프린터로, 용지 3장과 토너 20을 초기 값으로 설정해 동적으로 객체를 생성한다. 그 다음으로 showIJPinfo()와 showLPinfo() 함수를 호출하여 각각 잉크젯 프린터와 레이저 프린터의 정보를 출력한다. while 루프를 end 변수를 조건문으로 주어서 제어할 수 있도록 했다. 사용자에게 잉크젯(1) 또는 레이저(2) 프린터를 선택하고 int 타입의 what에 저장한다. 또한 프린트할 매수를 입력받도록 해 int 타입의 pages로 받는다. 이를 프린터 종류에 따라 처리할 수 있도록 if문을 사용했으며, 그 다음으로는 동적으로 할당했기 때문에 해당 프린터 객체의 포인터로 인쇄 종이 잔량(availableCount)을 리턴하는 멤버함수에 접근해 pages와 비교한다. pages가 해당 프린터의 availableCount보다 크다면 "용지가 부족하여 출력할 수 없습니다."를 출력하고 작다면 해당 프린터의 객체의 포인터로 printInkJet과 printLaser 멤버 함수에 접근해 프린트를 수행한다. 두 프린터의 객체의 포인터로 정보를 출력하는 멤버 함수에 접근해 정보를 출력하고 계속 프린트를 진행할 것인지를 입력받아 n을 입력받으면 end를 0으로 설정해 while문을 빠져나가고 동적으로 할당한 객체들을 delete 연산자로 메모리를 해제한다. y를 입력받으면 while문을 계속 돌도록 한다.

#아이디어 평가

(1) Printer 클래스에서 멤버 함수를 이용해서 호출될 때마다 pages 매수를 사용할 수 있었다.

(2) Printer 클래스를 상속받아서 InkJetPrinter 클래스는 잉크 잔량에 대해 작업할 수 있었고, LaserPrinter 클래스는 토너의 잔량을 관리하면서 인쇄하는 작업을 할 수 있었다. 또한, 상속을 이용하여 중복된 코드를 줄여 유연성을 높였다.

#문제를 해결한 키 아이디어 또는 알고리즘 설명

(1) 프린터 잔량 관리 코드가 핵심 알고리즘이라고 생각한다. printInkJet과 printLaser 함수는 각각 프린터의 잉크와 토너 잔량을 확인하여 인쇄 여부를 판단하는데 Printer 클래스의 print 함수는 availableCount를 감소시키지만 InkJetPrinter와 LaserPrinter는 잉크와 토너 잔량을 개별적으로 관리해서 잔량 부족할 때 인쇄가 불가능하도록 만들었다. 예를 들어서 printInkJet 함수에서는 print 함수를 호출한 뒤 잉크 잔량을 소모하며, 레이저 프린터의 경우 printLaser 함수에서 availableToner를 페이지에 비례해 소모하는 구조다. 이를 통해 잉크와 토너 관리를 각각할 수 있으며 프린터 사용자에게 잔량이 부족하다고 경고를 보낼 수 있다.

(2) main 함수에서 반복문을 이용하여 코드를 제어한 부분 또한 문제를 해결한 핵심 알고리즘이라고 생각한다. 각각의 프린터마다 getavailableCount를 통해 종이 잔량을 확인하고, 잔량이 충분하면 printInkJet나 printLaser 함수를 호출한다. 이를 통해 사용자에게 남은 용지 매수와 잉크와 토너의 잔량을 실시간으로 확인시키고, 인쇄할 때마다 상태를 갱신해 다음 인쇄가 가능한지를 알려준다.