

객체지향프로그래밍 과제#4

소프트웨어전공 202284012 김주원 / 소프트웨어전공 202304012 김도연

1. 소스 수행 결과 화면

```
Microsoft Visual Studio 디버그 × + v
현재 작동 중인 2대의 프린터는 아래와 같다
잉크젯 : Officejet V40 , HP , 남은 종이 5장 , 남은 잉크 10
레이저 : SCX-6x45 , 삼성전자 , 남은 종이 3장 , 남은 토너 20

프린터 (1: 잉크젯 , 2: 레이저)와 매수 입력>> 1 4
프린트하였습니다.
Officejet V40 , HP , 남은 종이 1장 , 남은 잉크 6
SCX-6x45 , 삼성전자 , 남은 종이 3장 , 남은 토너 20
계속 프린트 하시겠습니까(y/n)? y

프린터 (1: 잉크젯 , 2: 레이저)와 매수 입력>> 2 10
용지가 부족하여 프린트할 수 없습니다.
Officejet V40 , HP , 남은 종이 1장 , 남은 잉크 6
SCX-6x45 , 삼성전자 , 남은 종이 3장 , 남은 토너 20
계속 프린트 하시겠습니까(y/n)? y

프린터 (1: 잉크젯 , 2: 레이저)와 매수 입력>> 2 2
프린트하였습니다.
Officejet V40 , HP , 남은 종이 1장 , 남은 잉크 6
SCX-6x45 , 삼성전자 , 남은 종이 1장 , 남은 토너 18
계속 프린트 하시겠습니까(y/n)? n
```

2. 문제 정의

이 문제는 기본 프린터 클래스를 설계하고 이를 상속하여 잉크젯 프린터와 레이저 프린터 클래스를 구현하는 것이다. 모든 프린터는 모델명, 제조사, 인쇄 매수, 인쇄 종이 잔량을 가지며 `print(int pages)` 함수를 통해 인쇄할 수 있다. 잉크젯 프린터는 잉크 잔량, 레이저 프린터는 토너 잔량을 추가적으로 관리하며 각 프린터는 해당 자원이 충분할 때만 인쇄 기능을 수행한다. 잉크 또는 토너와 종이가 부족할 경우 경고 메시지를 출력하며 인쇄가 중단된다. 프로그램은 잉크젯과 레이저 프린터 객체를 각각 하나씩 동적으로 생성하고 인쇄 기능 실행 후 상태를 출력하여 자원 소모와 상태 변화를 확인하도록 구성된다.

3. 문제 해결 방법

<Printer 클래스 선언부 – Printer.h>

1. 기본 로직

Printer, InkjetPrinter, LaserPrinter 클래스를 정의하며 각 클래스는 프린터의 모델명,

제조사, 인쇄 매수, 잉크/토너/종이의 잔량을 속성으로 가진다. Printer 클래스는 print 함수와 showStatus 함수의 선언을 포함하고 있으며 print 함수는 순수 가상 함수로 선언되어 상속받은 클래스에서 반드시 구현해야 한다. InkjetPrinter와 LaserPrinter 클래스는 각각 잉크와 토너 잔량을 관리하는 속성을 추가하고 이를 통해 프린터의 상태를 처리할 수 있도록 설계된다.

2. 헤더 파일

#ifndef PRINTER_H, #define PRINTER_H, #endif를 사용하여 구성된 헤더 가드는 헤더 파일이 여러 번 포함되더라도 컴파일 에러가 발생하지 않도록 방지한다.

3. 함수 구현

Printer 클래스

- 생성자 Printer(const string& model, const string& manufacturer, int availableCount);
이 생성자는 Printer 객체를 생성할 때 모델명, 제조사, 남은 종이 수를 초기화한다. 이를 통해 **각 프린터 객체의 기본 속성을 설정**한다. printedCount는 기본값으로 0으로 설정된다.
- virtual void print(int pages) = 0;
print 함수는 **순수 가상 함수로 선언**되어 Printer 클래스에서 이 함수는 구현되지 않으며 Printer 클래스를 상속받은 자식 클래스에서 반드시 구현해야 한다. 이 함수는 인쇄할 페이지 수(pages)를 받아 해당 프린터가 인쇄를 할 수 있는지 확인하고 자원을 차감하는 등의 동작을 처리한다.
- virtual void showStatus() const;
showStatus 함수는 **Printer 클래스의 상태를 출력하는 함수**이다. 기본적으로 모델명, 제조사, 남은 종이 수를 출력하는 기능을 한다. 이 함수는 가상 함수이기 때문에 자식 클래스에서 잉크, 토너와 같은 추가적인 자원 정보를 출력하는 기능을 추가할 수 있다.

InkjetPrinter 클래스

- 생성자 InkjetPrinter(const string& model, const string& manufacturer, int availableCount, int availableInk);
Printer 클래스의 생성자를 호출하여 **기본적인 속성**(모델명, 제조사, 남은 종이 수)을 **설정**하고 추가적으로 **잉크 잔량을 설정**한다.

- `void print(int pages) override;`
print 함수는 Printer 클래스에서 상속받은 순수 가상 함수인 `print`를 오버라이드 하여 잉크와 종이가 충분한지 검사한 후 인쇄를 진행한다.
- `void showStatus() const override;`
showStatus 함수는 부모 클래스의 showStatus를 호출하여 기본 상태를 출력한 후 잉크 잔량도 함께 출력하는 추가적인 동작을 한다.

LaserPrinter 클래스

- 생성자 `LaserPrinter(const string& model, const string& manufacturer, int availableCount, int availableToner);`
Printer 클래스의 생성자를 호출하여 기본적인 속성을 설정하고 추가적으로 토너 잔량을 설정한다.
- `void print(int pages) override;`
print 함수는 Printer 클래스에서 상속받은 순수 가상 함수인 `print`를 오버라이드 하여 토너와 종이가 충분한지 검사한 후 인쇄를 진행한다.
- `void showStatus() const override;`
showStatus 함수는 부모 클래스의 showStatus를 호출하여 기본 상태를 출력하고 남은 토너 잔량도 함께 출력하는 기능을 추가한다.

4. 헤더 파일 보호

헤더 파일이 여러 번 포함되는 것을 방지하고 코드의 안정성과 가독성을 높이기 위해 `#endif` 사용한다.

<Printer 클래스 구현부 – Printer.cpp>

1. 기본 로직

헤더 파일에서 선언된 클래스들의 생성자와 함수들을 구현한다. 프린터의 기본 기능을 구현하는 Printer 클래스는 모델, 제조사, 잉크젯/레이저 프린터의 공통된 속성을 관리하고 `print`와 `showStatus`라는 공통적인 인터페이스를 정의한다. InkjetPrinter와 LaserPrinter는 Printer 클래스를 상속받아 각자 필요한 자원(잉크, 토너)을 관리하고 인쇄할 수 있는지 확인한 후 상태를 출력하는 방식으로 기능을 구현한다. `print` 함수에서는 인쇄가 가능한지 확인한 후 자원 부족 시 인쇄를 하지 않도록 경고 메시지를 출력한다. 각 프린터 클래스의 `showStatus` 함수는 잉크 또는 토너 잔량을 추가로 출력하여 상태를 더 자세히 보여준다.

2. 헤더 파일

#include "Printer.h"를 통해 Printer.h에서 선언된 Printer 클래스의 정의를 포함하여 해당 클래스의 멤버 변수와 함수들에 접근할 수 있게 한다.

#include <iostream>으로 cout, cin과 같은 표준 입출력 객체를 사용하기 위한 라이브러리 포함한다.

using namespace std;를 통해 std 네임스페이스에 있는 식별자를 쉽게 사용하도록 설정한다.

3. 함수 구현

Printer 클래스

- 생성자 `Printer::Printer(const string& model, const string& manufacturer, int availableCount)`
프린터 객체가 생성될 때 모델명, 제조사, 종이 잔량을 초기화한다. 또한 인쇄된 페이지 수는 0으로 초기화된다.
- `showStatus` 함수
기본적으로 **모델명, 제조사, 남은 종이 수를 출력**하는데 이를 부모 클래스에서 제공하며 자식 클래스에서는 **추가 자원(잉크/토너)을 출력할 수 있다.**

InkjetPrinter 클래스

- 생성자 `InkjetPrinter::InkjetPrinter(const string& model, const string& manufacturer, int availableCount, int availableInk)`
Printer의 생성자를 호출하여 기본 정보를 설정한 후 **잉크 잔량**을 초기화한다.
- `print` 함수
인쇄할 페이지 수만큼 **종이와 잉크 자원이 충분한지 검사한 후 자원이 충분하면 인쇄를 진행하고 자원을 차감**하며 "프린트하였습니다."라는 **메시지를 출력**한다. 만약 자원이 부족하면 "용지가 부족하여 프린트할 수 없습니다."라는 메시지를 출력한다.
- `showStatus` 함수
부모 클래스인 Printer의 `showStatus`를 호출하여 기본 상태를 출력한 후 **잉크 잔량을 추가로 출력**한다.

LaserPrinter 클래스

- 생성자 `LaserPrinter::LaserPrinter(const string& model, const string&`

manufacturer, int availableCount, int availableToner)

Printer 클래스의 생성자를 호출하여 기본 정보를 설정하고 **토너 잔량**을 초기화한다.

- **print 함수**

InkJetPrinter와 유사하게 종이와 토너가 충분한지 확인한 후 인쇄를 수행한다. 자원이 충분하면 인쇄하고 자원을 차감하며 "프린트하였습니다."를 출력하고 부족하면 "용지가 부족하여 프린트할 수 없습니다."라는 메시지를 출력한다.

- **showStatus 함수**

부모 클래스인 Printer의 showStatus를 호출하여 기본 상태를 출력하고 추가적으로 **남은 토너 잔량을 출력**한다.

<main 함수 – main.cpp>

1. 기본 로직

Printer.h를 포함하여 InkJetPrinter와 LaserPrinter 객체를 생성하고 이들을 사용하여 프린터 상태를 보여주고 프린트 요청을 처리한다. 사용자 입력을 통해 프린트 작업을 수행하고 각 프린터의 잔여 용지 및 잉크, 토너 상태를 보여준다. 사용자가 'n'을 입력하면 루프가 종료되고 프로그램이 끝난다.

2. 헤더 파일

#include "Printer.h"를 통해 Printer.h에서 선언된 Printer 클래스의 정의를 포함하여 해당 클래스의 멤버 변수와 함수들에 접근할 수 있게 한다.

#include <iostream>으로 cout, cin과 같은 표준 입출력 객체를 사용하기 위한 라이브러리 포함한다.

using namespace std;를 통해 std 네임스페이스에 있는 식별자를 쉽게 사용하도록 설정한다.

3. 함수 구현

InkJetPrinter와 LaserPrinter 객체 생성

- InkJetPrinter와 LaserPrinter 생성자에서 **Printer 클래스 생성자를 호출**하여 model, manufacturer, availableCount를 초기화함으로써 Printer 클래스에서 공통

으로 필요한 속성을 부모 클래스에서 한 번에 설정할 수 있게 한다.

- 각각의 자식 클래스에서 고유 멤버(InkjetPrinter의 availableInk와 LaserPrinter의 availableToner)는 자식 생성자에서 추가적으로 초기화된다.
- InkJetPrinter는 모델명이 "Officejet V40", 제조사가 "HP", 용지 5장, 잉크 10으로, LaserPrinter는 모델명이 "SCX-6x45", 제조사가 "삼성전자", 용지 3장, 토너 20으로 초기화된다.

생성한 두 프린터의 초기 상태 출력

- `inkjet.showStatus()`와 `laser.showStatus()`로 각 프린터 상태를 화면에 출력한다.
- 각 메서드들은 부모 클래스 `Printer`에서 상속받은 `showPrinter()`를 호출한 후 자식 클래스에서 고유 속성인 잉크 또는 토너와 같은 추가 정보를 출력하도록 오버라이딩 되어있다.
- 상속을 통해 `showPrinter()` 함수는 두 클래스에서 중복 코드 없이 공통된 기능을 처리하고 자식 클래스에서만 필요한 추가 정보를 출력하는 방식으로 재사용한다.

변수 선언

- `printerType` : 사용자가 인쇄할 프린터 유형을 선택하는 값
- `pages` : 인쇄할 페이지 수
- `continuePrinting` : 추가 인쇄 여부를 결정하기 위해 사용자가 y 또는 n으로 응답한 값 저장

프린터 유형과 페이지 수 입력

- `do-while` 루프로 프로그램이 종료될 때까지 계속 반복한다.
- `printerType`이 1인 경우 `inkjet.print(pages)`를 통해 잉크젯 프린터로 인쇄하고 `printerType`이 2인 경우 `laser.print(pages)`를 통해 레이저 프린터로 인쇄하도록 한다.
- 다른 값이 입력되면 "잘못된 프린터 선택입니다." 라는 메시지를 출력한다.

인쇄 후 상태 출력

- 인쇄 작업이 끝난 후 `inkjet.showStatus();`와 `laser.showStatus();`를 이용하여 인쇄가 완료된 이후의 각 프린터의 남은 용지와 잉크, 토너 상태를 다시 보여준다.

프로그램 종료

- 인쇄 작업 후 사용자가 계속 프린트를 하고 싶은지 묻는 메시지 출력한다.
- `continuePrinting`에 사용자 입력을 받아 'y'가 입력되면 루프가 계속되어 새로운 인쇄 작업을 받고 'n'이 입력되면 while 루프가 종료되어 프로그램을 종료한다.