

# Print.h 소스코드 분석

```
10주차 과제 Print
1 #ifndef Print_H
2 #define Print_H
3
4 class Print {
5     string model, manufacturer;
6     int pcount, availableCount;
7 protected:
8     Print(string model, string manufacturer, int availableCount) {
9         this->model = model;
10        this->manufacturer = manufacturer;
11        this->availableCount = availableCount;
12    }
13    bool print(int pages) {
14        if (availableCount < pages) {
15            cout << "용지가 부족하여 프린트 할 수 없습니다.\n";
16            return false;
17        }
18        for (int i = 0; i < pages; i++) {
19            availableCount--;
20        }
21        return true;
22    }
23    string get_model() { return model; }
24    string get_manufacturer() { return manufacturer; }
25    int get_availableCount() { return availableCount; }
26 };
27
28 #endif
```

# Print.h 소스코드 분석

1, 2, 28번 라인에 `#ifndef`, `#define`, `#endif` 명령어를 통해 헤더파일이 중복으로 정의되는 것을 방지합니다.

4~6번 라인에 `Print`클래스의 속성 `model`, `manufacturer`, `pcount`, `availableCount`를 선언합니다.

7~26번 라인에서는 접근지정자 `protected`로 상속받는 클래스에서 접근이 가능하도록 하며, 생성자 `Print(string model, string manufacturer, int availableCount)`로 `model`, `manufacturer`, `availableCount`는 각각 인자로 받은 값을 해당 변수값에 저장합니다.

멤버함수 `bool print(int pages)`에서는 프린트할 페이지 수가 남은 종이 수보다 많은 경우 "용지가 부족"하다는 메시지를 출력하고 `false`를 반환합니다. 충분한 경우에는 `availableCount`를 감소시키고 `true`를 반환합니다.

또한 `get_model()`, `get_manufacturer()`, `get_availableCount()`를 이용하여 각각의 값을 반환합니다.

# PrintInkJet.h 소스코드 분석

```
10주차 과제 (전역 범위)

1 #ifndef PrintInkJet_H
2 #define PrintInkJet_H
3
4 class PrintInkJet : public Print {
5     int availableInk;
6 public:
7     PrintInkJet(string model, string manufacturer, int availableCount, int availableInk) : Print(model, manufacturer, availableCount) {
8         this->availableInk = availableInk;
9     }
10    bool printInkJet(int pages) {
11        if (print(pages));
12        else return false;
13        if (availableInk < pages) {
14            cout << "잉크가 부족하여 프린트 할 수 없습니다.\n";
15            return false;
16        }
17        for (int i = 0; i < pages; i++) {
18            availableInk--;
19        }
20        return true;
21    }
22    void show() {
23        cout << get_model() << "\t," << get_manufacturer() << "\t,남은 종이 " << get_availableCount() << "장\t,남은 잉크 " << availableInk << endl;
24    }
25 };
26
27 #endif
```

# PrintInkJet.h 소스코드 분석

1, 2, 27번 라인에 `#ifndef`, `#define`, `#endif` 명령어를 통해 헤더파일이 중복으로 정의되는 것을 방지합니다.

4~25번 라인에서 `PrintInkJet` 클래스는 `Print` 클래스를 상속받아서 구성되며, 속성 `availableInk`를 선언하고 `Print`의 생성자를 호출하여 `model`, `manufacturer`, `availableCount`의 값을 초기화 합니다.

멤버함수 `bool printInkJet(int pages)`에서는 `print(pages)` 메서드를 호출하여 종이 부족 여부를 먼저 확인합니다.

그 후 잉크 수가 부족하면 잉크가 부족하다는 메시지를 출력하고 `false`를 반환합니다.

충분한 경우에는 `availableInk`를 페이지 수만큼 감소시키고 `true`를 반환합니다.

그리고 `void show()`함수를 이용하여 프린터의 모델명, 제조사, 남은 종이 수, 남은 잉크 수를 출력합니다.

# PrintLaser.h 소스코드 분석

```
1 #ifndef PrintLaser_H
2 #define PrintLaser_H
3
4 class PrintLaser : public Print {
5     int availableToner;
6 public:
7     PrintLaser(string model, string manufacturer, int availableCount, int availableToner) : Print(model, manufacturer, availableCount) {
8         this->availableToner = availableToner;
9     }
10    bool printLaser(int pages) {
11        if (print(pages));
12        else return false;
13        if (availableToner < pages) {
14            cout << "토너가 부족하여 프린트 할 수 없습니다.\n";
15            return false;
16        }
17        for (int i = 0; i < pages; i++) {
18            availableToner--;
19        }
20        return true;
21    }
22    void show() {
23        cout << get_model() << " , " << get_manufacturer() << "\t, 남은 종이 " << get_availableCount() << "장\t, 남은 토너 " << availableToner << endl;
24    }
25 };
26
27 #endif
```

# PrintLaser.h 소스코드 분석

1, 2, 27번 라인에 `#ifndef`, `#define`, `#endif` 명령어를 통해 헤더파일이 중복으로 정의되는 것을 방지합니다.

4~25번 라인에서 `PrintLaser`클래스는 `Print` 클래스를 상속받아서 구성되며, 속성 `availableToner`를 선언하고 `Print`의 생성자를 호출하여 `model`, `manufacturer`, `availableCount`의 값을 초기화 합니다.

멤버함수 `bool printLaser(int pages)`에서는 `print(pages)` 메서드를 호출하여 종이 부족 여부를 먼저 확인합니다.

그 후 토너 수가 부족하면 토너가 부족하다는 메시지를 출력하고 `false`를 반환합니다.

충분한 경우에는 `availableToner`를 페이지 수만큼 감소시키고 `true`를 반환합니다. 그리고 `void show()`함수를 이용하여 프린터의 모델명, 제조사, 남은 종이 수, 남은 토너 수를 출력합니다.

# Main.cpp 소스코드 분석

```
10주차 과제 (전역 범위) main()
1  #include<iostream>
2  using namespace std;
3
4  #include "Print.h"
5  #include "PrintInkJet.h"
6  #include "PrintLaser.h"
7
8  int main() {
9      int pnum, pages;
10     char yon;
11     PrintInkJet* inkjet = new PrintInkJet("Officejet V40", "Hp", 5, 10);
12     PrintLaser* laser = new PrintLaser("SCX-6x45", "삼성전자", 3, 20);
13     cout << "현재 작동중인 2 대의 프린터는 아래와 같다\n";
14     cout << "잉크젯 : ";
15     inkjet->show();
16     cout << "레이저 : ";
17     laser->show();
18     cout << endl;
19
20     while (true) {
21         cout << "프린터(1:잉크젯, 2:레이저)와 매수 입력>>";
22         cin >> pnum >> pages;
23         if (pnum == 1)
24             if (inkjet->printInkJet(pages))
25                 cout << "프린트 하였습니다.\n";
26         if (pnum == 2)
27             if (laser->printLaser(pages))
28                 cout << "프린트 하였습니다.\n";
29         if (pnum != 1 && pnum != 2) cout << "프린터를 잘못 선택하셨습니다.\n";
30         inkjet->show();
31         laser->show();
32         while (true) {
33             cout << "계속 프린트 하시겠습니까(y/n)>>";
34             cin >> yon;
35             cout << endl;
36             if (yon == 'n')
37                 return 0;
38             else if (yon == 'y')
39                 break;
40             else
41                 cout << "잘못 입력하셨습니다.\n";
42         }
43     }
44     delete inkjet;
45     delete laser;
46 }
```

# Main.cpp 소스코드 분석

4~6번 라인에 "Print.h", "PrintInkJet.h", "PrintLaser.h" 3개의 헤더파일을 include 합니다.

8~46번 라인의 메인함수에서는 프린터의 종류를 선택할 pnum, 프린트할 장수를 나타낼 pages, 프린트의 지속 여부를 결정할 yon 변수들을 선언하고, PrintInkJet과 PrintLaser의 객체 inkjet과 laser를 각각 생성하고 프로그램 시작 시 각각의 프린터에 대한 상태 정보를 출력합니다.

그 후 while 루프를 통해 사용자가 입력한 프린터 번호와 매수에 따라 프린트를 진행합니다.

만약 pnum 값이 1이면 inkjet->printInkJet(pages)를 호출하여 잉크젯 프린터로 출력하고, pnum 값이 2이면 laser->printLaser(pages)를 호출하여 레이저 프린터로 출력합니다.

출력 성공 시 "프린트 하였습니다." 메시지를 출력합니다. 또한 pnum이 1, 2 외의 값일 경우 "프린터를 잘못 선택하셨습니다."를 출력합니다.

프린트 후에는 show() 메서드를 호출하여 각 프린터의 남은 종이 및 잉크/토너 상태를 출력합니다.

그 후에 yon 값을 입력받아 y일 경우 루프를 반복하고, n일 경우 프로그램을 종료합니다. y나 n이 아닌 다른 값이 입력되면 "잘못 입력하셨습니다." 메시지를 출력하고 다시 묻습니다.

마지막으로 프로그램 종료 시에 동적 할당된 inkjet과 laser 객체를 삭제합니다.