

문제 정의

이 코딩은 두 종류의 프린터, 잉크젯 프린터와 레이저 프린터를 사용자로부터 인쇄할 매수를 입력받고, 해당 매수를 인쇄할 수 있는지를 판단해 인쇄한다. 각 프린터는 인쇄 때마다 잉크 또는 토너와 종이 잔량을 소비하며, 부족할 경우 메시지를 표시한다. 인쇄가 완료되면, 남은 종이 및 잉크 또는 토너의 상태를 출력한다.

문제 해결 방법

1. Printer라는 기본 클래스가 인쇄 관련 기본 기능, InkJetPrinter와 LaserPrinter가 Printer 클래스를 상속받아 잉크 및 토너 관리

```
5  class Printer {
6      protected:
7          string model; // 모델명
8          string manufacturer; // 제조사
9          int printedCount; // 인쇄 매수
10         int availableCount; // 인쇄 종이 잔량
11
12     public: // 외부에서 접근 가능한 멤버
13         Printer(string model, string manufacturer, int availableCount) {
14             this->model = model;
15             this->manufacturer = manufacturer;
16             this->printedCount = 0;
17             this->availableCount = availableCount;
18         }
19
20         bool print(int pages) { // 인쇄 시도 함수, 인쇄 성공시 printedCount, availableCount 갱신
21             if (pages > availableCount) {
22                 cout << "용지가 부족하여 프린트할 수 없습니다.\n";
23                 return false; // 페이지 수가 현재 종이 잔량보다 많으면 bool에서 false 출력
24             }
25             printedCount += pages; // 인쇄된 페이지 추가(
26             availableCount -= pages; // 인쇄 성공 시 페이지만큼 종이 줄이기
27             return true; // 성공 시 true 출력
```

(기본 클래스 Printer)

그 다음 이것을 상속받는 잉크젯 프린터와 레이저 프린터를 기본 클래스 생성자를 통해 초기화한다.

```

38 class InkJetPrinter : public Printer { // Printer 클래스를 상속받기
39     int availableInk; // 잉크 잔량
40
41 public:
42     InkJetPrinter(string model, string manufacturer, int availableCount, int availableInk)
43         : Printer(model, manufacturer, availableCount) {
44         this->availableInk = availableInk; // 기본 클래스 Printer 생성자를 통해 초기화
45     }
46
47     ~InkJetPrinter(){
48         cout << "InkJetPrinter 소멸자 호출: " << model << endl;
49     }
50
51     bool print(int pages) { // print 함수 재정의
52         if (Printer::print(pages)) { // 부모클래스 print 함수 호출하여 종이 잔량 확인하는 방법
53             if (pages > availableInk) {
54                 cout << "잉크가 부족하여 프린트할 수 없습니다.\n";
55                 availableCount += pages;
56                 return false;
57             }
58             availableInk -= pages;
59             return true;
60         }
61         return false;
62     }
63
64     void showStatus() const { // showStatus 함수 재정의
65         Printer::showStatus();
66         cout << ", 남은 잉크 " << availableInk << endl;
67     }
68 };

```

(상속받은 클래스 InkJetPrinter)

같은 방식으로 레이저프린터도 작성해준다

```

70 class LaserPrinter : public Printer {
71     int availableToner; // 레이저 프린터 클래스의 토너 잔량
72
73 public:
74     LaserPrinter(string model, string manufacturer, int availableCount, int availableToner)
75         : Printer(model, manufacturer, availableCount) {
76         this->availableToner = availableToner;
77     }
78
79     ~LaserPrinter() {
80         cout << "LaserPrinter 소멸자 호출: " << model << endl;
81     }
82
83     bool print(int pages) {
84         if (Printer::print(pages)) {
85             if (pages > availableToner) {
86                 cout << "토너가 부족하여 프린트할 수 없습니다.\n";
87                 availableCount += pages; // 사용된 종이 수 복구하기
88                 return false;
89             }
90             availableToner -= pages; // 인쇄시 토너 잔량 줄이기
91             return true;
92         }
93         return false;
94     }
95
96     void showStatus() const {
97         Printer::showStatus();
98         cout << ", 남은 토너 " << availableToner << endl;
99     }
100 };

```

(상속받은 클래스 LaserPrinter)

2. 사용자가 입력한 매수가 남아있는 종이 잔량 또는 잉크/토너 잔량보다 많을 경우, 인쇄가 불가능하다는 메시지를 출력하고, 종이 잔량을 복구하는 방법을 사용한다.

```
23 bool print(int pages) { // 인쇄 시도 함수, 인쇄 성공시 printedCount, availableCount 갱신
24     if (pages > availableCount) {
25         cout << "용지가 부족하여 프린트할 수 없습니다.\n";
26         return false; // 페이지 수가 현재 종이 잔량보다 많으면 bool에서 false 출력
27     }
28     printedCount += pages; // 인쇄된 페이지 추가
29     availableCount -= pages; // 인쇄 성공 시 페이지만큼 종이 줄이기
30     return true; // 성공 시 true 출력
31 }
32
51 bool print(int pages) { // print 함수 재정의
52     if (Printer::print(pages)) { // 부모클래스 print 함수 호출하여 종이 잔량 확인하는 방법
53         if (pages > availableInk) {
54             cout << "잉크가 부족하여 프린트할 수 없습니다.\n";
55             availableCount += pages;
56             return false;
57         }
58         availableInk -= pages;
59         return true;
60     }
61     return false;
62 }
63
82 bool print(int pages) {
83     if (Printer::print(pages)) {
84         if (pages > availableToner) {
85             cout << "토너가 부족하여 프린트할 수 없습니다.\n";
86             availableCount += pages; // 사용된 종이 수 복구하기
87             return false;
88         }
89         availableToner -= pages; // 인쇄시 토너 잔량 줄이기
90         return true;
91     }
92     return false;
93 }
```

3. 인쇄 작업을 반복할 수 있도록 while 반복문과 'y/n'의 입력을 통해 반복을 제어한다.

```

115     while (yesorno == 'y') {
116         cout << "프린터(1:잉크젯, 2:레이저)와 매수 입력>>";
117         cin >> printertype >> pages;
118
119         if (printertype == 1) {
120             if (inkjet.print(pages)) {
121                 cout << "프린트하였습니다.\n";
122             }
123             inkjet.showStatus();
124         }
125         else if (printertype == 2) {
126             if (laser.print(pages)) {
127                 cout << "프린트하였습니다.\n";
128             }
129             laser.showStatus();
130         }
131         else {
132             cout << "잘못된 프린터 선택입니다.\n";
133         }
134         cout << endl;
135         cout << "계속 프린트 하시겠습니까?(y/n)>>";
136         cin >> yesorno;
137     }

```

아이디어 평가

1. Printer 클래스를 사용하여 기본 인쇄 기능을 구현한 뒤, 이를 상속받아 잉크젯과 레이저 프린터에 특화된 기능을 추가한 아이디어는 코드 재사용성과 확장성 측면에서 효율적이다.
2. 인쇄할 매수와 잉크/토너 잔량을 모두 고려하여 인쇄가 불가능할 경우 사용자에게 메시지를 출력하고, 사용된 종이, 잉크, 토너 수를 다시 추가하는 방법은 이 문제 해결에 주요 아이디어가 되는 것 같다.
3. while문을 사용하여 인쇄 요청을 받는 구조이고, 프린트 성공 시 현재의 상태를 보여주는 showStatus()도 추가되어있다.

문제를 해결한 키 아이디어 또는 알고리즘 설명

이번 문제의 핵심은 "상속" 이라고 볼 수 있다. 상속받는 형태의 코드는 확장성을 높여, 새로운 프린터 종류를 추가할 때 기본 `Printer` 클래스를 상속받아 필요한 기능만 재정의하면 되므로 전체 코드를 다시 작성하지 않고도 기능을 확장할 수 있게 한다.

또한, 요청된 프린트 페이지가 현재 보유한 잉크 및 토너보다 많을 시에는 부모 클래스 `print` 함수를 호출하여 잔량을 확인하고, `availableInk`, `availableToner`로 비교하여 먼저 실행된 `Printer` 클래스의 사용된 종이 수를 복구하는 방법은 매우 좋다고 생각한다.