객체지향프로그래밍 과제 #4

소프트웨어전공 202284011 김연재

소프트웨어전공 202004006 고창석

[문제 정의]

Printer 클래스와 두 개의 파생 클래스(InkJetPrinter, LaserPrinter)를 설계하며, 각 클래스에 적절한 접근 지정자로 멤버 변수와 함수, 생성자, 소멸자를 작성해야 한다.

InkJetPrinter와 LaserPrinter 객체는 동적으로 생성하며, 프로그램이 종료될 때까지 지속적으로 인쇄 작업을 수행할 수 있어야 한다.

각 프린터의 남은 용지, 잉크 또는 토너 상태를 실시간으로 확인하고 부족할 경우 적절한 메시지를 출력해야 한다.

[문제 해결 방법]

1. 클래스 설계

```
5   class Printer {
6       string model;
7       string manufacturer;
8       int printedCount;
9       int availableCount;
10  protected:
11      Printer(string mo = "", string me = "", int a = 0) {
12          this->model = mo;
13          this->manufacturer = me;
14          this->availableCount = a;
15          this->printedCount = 0;
16      }
17      bool isValidPrint(int pages) {
18          if (availableCount >= pages) return true;
19          else return false;
20      }
21      void print(int pages) {
22          if (isValidPrint(pages)) {
23              printedCount += pages;
24              availableCount -= pages;
25          }
26          else cout << "용지가 부족하여 프린트할 수 없습니다." << endl;
27      }
28      void showPrinter() { cout << model << ", " << manufacturer << ", 남은 종이 " <<
29          availableCount << "장"; }
30  };
```

Printer 클래스

공통 속성: 프린터 모델명, 제조사, 인쇄한 매수(printedCount), 남은 용지 (availableCount)를 포함.

접근 지정자: 공통 속성은 protected로 지정하여 파생 클래스에서 접근할 수 있도록 설정하고, 필요한 공통 메서드(isValidPrint, print, showPrinter)를 제공.

생성자 및 소멸자: 생성자는 파생 클래스에서 모델명, 제조사, 용지 수를 설정할 수 있도록 하며, 소멸자는 명시적으로 작성하지 않고 기본 소멸자를 사용.

```
32  class InkJetPrinter : public Printer {
33      int availableInk;
34  public:
35      InkJetPrinter(string mo = "", string me = "", int a = 0, int i = 0) : Printer(mo, me, a)
36      { this->availableInk = i; }
37      bool isValidInkJetPrint(int pages) {
38          if (availableInk > pages) return true;
39          else return false;
40      }
41      void printInkJet(int pages) {
42          if (isValidPrint(pages)) {
43              if (isValidInkJetPrint(pages)) {
44                  print(pages);
45                  availableInk -= pages;
46                  cout << "프린트하였습니다" << endl;
47              }
48              else {
49                  cout << "잉크가 부족하여 프린트할 수 없습니다." << endl;
50              }
51          }
52          else cout << "용지가 부족하여 프린트할 수 없습니다." << endl;
53      }
54      void showInkJetPrinter() {
55          showPrinter();
56          cout << ", 남은 토너 " << availableInk << endl;
57      }
58  };
```

InkJetPrinter 클래스

추가 속성: 잉크 잔량(availableInk)을 추가하여 Printer 클래스를 상속받아 잉크젯 프린터의 특성을 반영.

메서드:

isValidInkJetPrint: 주어진 인쇄 매수에 대한 잉크 잔량이 충분한지 확인.

printInkJet: 잉크와 용지가 충분할 때 인쇄 작업을 수행.

showInkJetPrinter: 잉크젯 프린터의 모델명, 제조사, 남은 용지, 남은 잉크를 출력.

```
class LaserPrinter : public Printer {
    int availableToner;
public:
    LaserPrinter(string mo = "", string me = "", int a = 0, int t = 0) : Printer(mo, me, a)
    { this->availableToner = t; }
    bool isValidLaserPrint(int pages) {
        if (availableToner > pages) return true;
        else return false;
    }
    void printInkJet(int pages) {
        if (isValidPrint(pages)) {
            if (isValidLaserPrint(pages)) {
                print(pages);
                availableToner -= pages;
                cout << "프린트하였습니다" << endl;
            }
            else {
                cout << "토너가 부족하여 프린트할 수 없습니다." << endl;
            }
        }
        else cout << "용지가 부족하여 프린트할 수 없습니다." << endl;
    }
    void showLaserPrinter() {
        showPrinter();
        cout << ", 남은 토너 " << availableToner << endl;
    }
};
```

LaserPrinter 클래스

추가 속성: 토너 잔량(availableToner)을 추가하여 레이저 프린터의 특성을 반영.

메서드:

isValidLaserPrint: 주어진 인쇄 매수에 대한 토너 잔량이 충분한지 확인.

printLaser: 토너와 용지가 충분할 때 인쇄 작업을 수행.

showLaserPrinter: 레이저 프린터의 모델명, 제조사, 남은 용지, 남은 토너를 출력.

```
85  int main() {
86      InkJetPrinter ink("Officejet V40", "HP", 5, 10);
87      LaserPrinter laser("SCX-6x45", "삼성전자", 3, 20);
88
89      cout << "현재 작동중인 2대의 프린터는 아래와 같다" << endl;
90      cout << "잉크젯 : ";
91      ink.showInkJetPrinter();
92      cout << "레이저 : ";
93      laser.showLaserPrinter();
94
95      int printer, paper;
96      char ch;
97      while (true) {
98          cout << endl << "프린터(1:잉크젯, 2:레이저)와 매수 입력>>";
99          cin >> printer >> paper;
100         switch (printer) {
101         case 1: ink.printInkJet(paper); break;
102         case 2: laser.printInkJet(paper); break;
103         default: break;
104         }
105         ink.showInkJetPrinter();
106         laser.showLaserPrinter();
107
108         cout << "계속 프린트 하시겠습니까(y/n)>>";
109         cin >> ch;
110         if (ch == 'n') break;
111         else continue;
112     }
113
114     return 0;
115 }
```

2. main 함수 구성

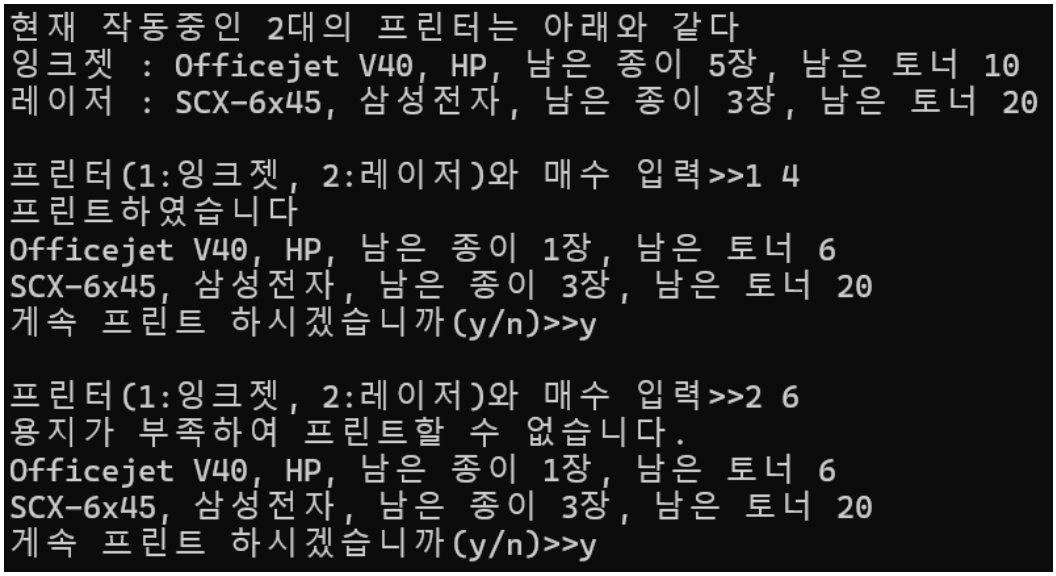
InkJetPrinter와 LaserPrinter 객체를 생성하고 초기 정보를 출력.

사용자로부터 프린터 종류(1: 잉크젯, 2: 레이저)와 인쇄할 페이지 수를 입력받아 인쇄 작업을 수행.

인쇄 후에는 각 프린터의 남은 용지, 잉크 또는 토너 상태를 출력.

인쇄를 계속할지 여부를 묻고, 'n'을 입력받으면 종료.

[소스 수행 결과 화면 캡처]



[평가]

김연재: 이번 과제를 통해 상속을 활용한 코드 재사용성의 장점을 경험할 수 있었다. 상위 클래스에 정의된 메서드를 하위 클래스에서 호출하여 공통 기능을 재사용하면서도 각 프린터의 고유 기능을 추가할 수 있었다. 특히 `showPrinter`와 같이 공통적인 출력 형식은 상위 클래스에서 정의해 두니 코드가 더욱 간결해지고 가독성이 높아졌다. 향후 코드를 작성할 때도 상속과 재사용을 염두에 두고 설계할 생각이다.

고창석: 잉크젯과 레이저 프린터가 공통적으로 가지고 있는 용지 수 체크나 인쇄 작업을 상위 클래스에 두고, 각각의 특징적인 요소(잉크와 토너)는 하위 클래스에서 별도로 구현할 수 있어서 객체지향의 장점을 느낄 수 있었다. 하지만 `printInkJet`와 `printLaser` 메서드가 비슷한 형태로 반복되는 부분이 있어 개선할 수 있는 방법이 있을 것 같다.