

과제4 소스코드 설명 - 서신우

문제 정의

- 우선 기본 클래스를 만들고 다른 2개의 프린터 클래스에서 기본 클래스를 상속받는다.
- 잉크 프린터와 레이저 프린터에서 해당 클래스에 맞는 멤버와 함수를 추가한다.
- main() 함수에서 잉크 프린터와 레이저 프린터를 생성하고 의도한 실행 결과가 나오도록 구현한다.

문제 해결

- 기본 클래스에는 상속받은 클래스에도 필요한 멤버와 함수들을 정의한다.
이때 main() 함수에서 2개의 프린터 클래스를 헤더파일로 가져올 때 중복되면 에러가 나니
`#ifndef _MAIN_H` 를 추가해준다.
- 잉크프린터 클래스와 레이저프린터 클래스에는 멤버에 값을 넣어줄 수 있도록 생성자를 구현해야한다.
또 잉크프린터 클래스에는 `availableInk` 멤버를, 레이저프린터 클래스에는 `availableToner` 멤버를 추가한다.
- `printInkJet` 과 `printLaser` 함수를 만들 때 `int` 타입의 파라미터를 받고, 받은 값만큼 남은 종이와 남은 잉크/토너 에서 값을 빼야한다. 만약 남은 종이 또는 남은 잉크/토너가 부족해서 프린트를 할 수 없는 경우 에러 메시지를 띄워주는 기능도 구현해야한다.

Code

Printer.h

Printer 클래스는 잉크프린터 클래스와 레이저프린터 클래스의 기본 클래스가 되기 때문에 공통되는 멤버 필드를 갖고 있다.

```
#ifndef _MAIN_H
#define _MAIN_H

class Printer {
public:
    string model;
    string manufacturer;
```

```

        int printedCount;
        int availableCount;
        void print(){ cout << "프린트하였습니다." << endl; };
};

#endif

```

PrinterInkJet.h

PrinterInkJet 클래스는 Printer 클래스를 상속받는데 public 을 통해 상속받아 기본 클래스의 protected, public 멤버들을 접근 지정 변경 없이 파생 클래스에 그대로 상속 확장받는다.

```

class PrinterInkJet : public Printer {
    // ~~~
}

```

PrinterInkJet 생성자는 인스턴스를 생성할 때 필드에 필요한 값들을 받아 초기화 시켜준다.

```

PrinterInkJet(string model, string manufacturer, int availableInk)
{
    this->model = model;
    this->manufacturer = manufacturer;
    this->availableCount = availableCount;
    this->availableInk = inkCount;
    this->printedCount = 0;
};

```

그리고 기본 클래스에서는 상속받지 못한, 잉크프린터 클래스만의 멤버 필드와 멤버 함수를 정의한다.

```

int availableInk;
void printInkJet(int pages){
    // ~~
}

```

printInkJet() 함수의 역할은 사용자가 프린트할 종이의 개수를 입력하면 현재 프린터기에 있는 종이의 수와 잉크 양이 부족하지 않은지 체크하고, 부족하지 않으면 프린트를 진행한다. 음 프린트하면서 사용한 종이와 잉크 양을 기존 양에서 빼주는 것이다. 만약, 종이 또는 잉크의 양이 부족해 프린트를 진행할 수 없다면 에러 메시지를 보여준다.

```
void printInkJet(int pages)
{
    if ((availableCount >= pages) && (availableInk >= pages))
    {
        if (pages != 0)
        {
            print();
        }
        availableCount -= pages;
        availableInk -= pages;
        printedCount++;
        cout << model + " " + manufacturer + " 남은 종이 " << a
    }
    else
    {
        cout << "용지 또는 잉크가 부족하여 프린트할 수 없습니다." << e
    };
};
}
```

PrinterLaser.h

PrinterInkJet 클래스와 PrinterLaser 클래스는 매우 유사하다.

main.cpp

우선 잉크프린터와 레이저프린터를 생성자를 통해 인스턴스를 동적으로 생성한다.

```
PrinterInkJet inkPrinter("Officejet V40", "HP", 5, 10);
PrinterLaser laserPrinter("SC-6x45", "삼성전자", 3, 20);
```

무한 반복을 위해 while 문을 사용한다.

```
while (true)
{
    // ~~~
}
```

사용자로부터 어떤 프린터를 사용할지, 프린트할 매수는 얼마인지를 입력받는다.
또, 프린트할 종이 수는 문자열이므로 int 타입으로 바꾸는 과정을 진행해준다.

```
cout << "프린터(1:잉크젯, 2:레이저)와 매수 입력 : ";
string input;
getline(cin, input);
int pages = stoi(input.substr(2));
```

사용자가 잉크프린트를 선택했다면 잉크프린터의 printInkJet() 함수를 실행시키고,
사용자가 레이저프린트를 선택했다면 레이저프린터의 printLaser() 함수를 실행시킨다.

```
if (input[0] == '1')
{
    inkPrinter.printInkJet(pages);
    laserPrinter.printLaser(0);
}

if (input[0] == '2')
{
    inkPrinter.printInkJet(0);
    laserPrinter.printLaser(pages);
}
```