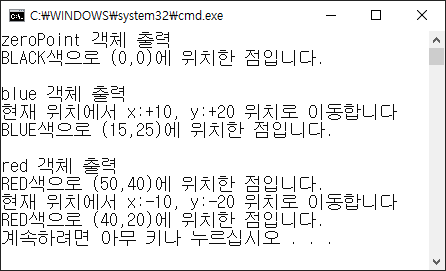
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2021\_1\_C++ \_11 | 학번 : | 20185309 | 이름 : | 황명원 |

* **프로그램 과제**

1. Point 클래스를 상속받는 ColorPoint 클래스를 작성하고 제시된 결과처럼 실행될 수 있도록 main()을 완성하시오.

class Point {

 int x, y;

public:

Point(int x, int y) { this->x = x; this->y = y; }

int getX() { return x; }

int getY() { return y; }

protected:

void move\_up(int x, int y) { this->x += x; this->y += y; }

void move\_down(int x, int y) { this->x -= x; this->y -= y; }

};

int main() {

ColorPoint zeroPoint; // BLACK에 (0, 0) 위치의 점

ColorPoint blue(5, 5);

cout << endl<<"blue 객체 출력" << endl;

cout << "현재 위치에서 x:+10, y:+20 위치로 이동합니다" << endl;

blue.setPoint(10, 20, '+');

ColorPoint red(50, 40, "RED");

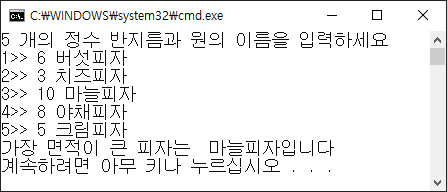
cout << "현재 위치에서 x:-10, y:-20 위치로 이동합니다" << endl;

red.setPoint(10, 20, '-');

}

|  |
| --- |
| [프로그램 소스]  #include <iostream> #include <string> using namespace std;  class Point {  int x, y; public:  Point(int x, int y) { this->x = x; this->y = y; }  int getX() { return x; }  int getY() { return y; } protected:  void move\_up(int x, int y) { this->x += x; this->y += y; }  void move\_down(int x, int y) { this->x -= x; this->y -= y; } }; class ColorPoint : public Point{ public:  ColorPoint() : Point(0,0){}  ColorPoint(int x,int y):Point(x,y){}  ColorPoint(int x,int y,string color):Point(x,y){}  void setPoint(int n1,int n2,char c); };  void ColorPoint::setPoint(int n1, int n2, char c) {  if(c=='+'){  move\_up(n1,n2);  }  else if(c=='-'){  move\_down(n1,n2);  } }  int main() {  ColorPoint zeroPoint; // BLACK에 (0, 0) 위치의 점   cout << endl<<"zeroPoint 객체 출력" << endl;  cout<<"BLACK색으로 (0,0)에 위치한 점입니다."<<endl;   ColorPoint blue(5, 5);  cout << endl<<"blue 객체 출력" << endl;  cout << "현재 위치에서 x:+10, y:+20 위치로 이동합니다" << endl;  blue.setPoint(10, 20, '+');  cout<<"BLUE색으로 ("<<blue.getX()<<","<<blue.getY()<<")에 위치한 점입니다."<<endl;   ColorPoint red(50, 40, "RED");  cout << endl<<"red 객체 출력" << endl;  cout<<"RED색으로 ("<<red.getX()<<","<<red.getY()<<")에 위치한 점입니다."<<endl;  cout << "현재 위치에서 x:-10, y:-20 위치로 이동합니다" << endl;  red.setPoint(10, 20, '-');  cout<<"RED색으로 ("<<red.getX()<<","<<red.getY()<<")에 위치한 점입니다."<<endl; } |
| [실행 결과]  텍스트이(가) 표시된 사진  자동 생성된 설명 |

1. 원을 추상화 한 Circle 클래스에 대하여 다음과 같이 배열을 선언하여 제시된 결과처럼 실행될 수 있도록 Circle을 상속받은 NamedCircle 클래스와 main()함수 등 필요한 함수를 작성하시오

class Circle {

int radius;

public:

Circle(int radius = 0) { this->radius = radius; }

int getRadius() { return radius; }

void setRadius(int radius) { this->radius = radius; }

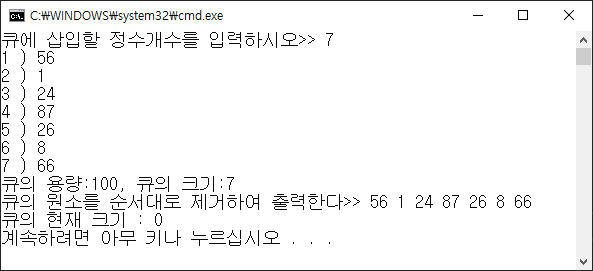
double getArea() { return 3.14\*radius\*radius; };

};

NamedCircle c[5];

|  |
| --- |
| [프로그램 소스]  #include <iostream> #include <string> using namespace std;  class Circle {  int radius; public:  Circle(int radius = 0) { this->radius = radius; }  int getRadius() { return radius; }  void setRadius(int radius) { this->radius = radius; }  double getArea() { return 3.14\*radius\*radius; }; }; class NamedCircle : public Circle{  string name; public:  void setRadiName(int r,string name){  setRadius(r);  this->name=name;  }  string getname(){  return name;  }  }; int main() {  string name;  int radius;  NamedCircle c[5];   cout<<"5 개의 정수 반지름과 원의 이름을 입력하세요"<<endl;  for(int i=0 ;i<5 ; i++){  cout<<i+1<<">>";  cin>>radius>>name;  c[i].setRadiName(radius,name);  }   int max=0;  int numOfmax=0;  for(int i=0 ; i<5 ;i++){  if(c[i].getArea()>max) {  max = c[i].getArea();  numOfmax=i;  }  }  cout<<"가장 면적이 큰 피자는 "<<c[numOfmax].getname()<<"입니다"; } |
| [실행 결과]  텍스트이(가) 표시된 사진  자동 생성된 설명 |

1. BaseArray클래스를 상속 받아 큐처럼 작동하는 MyQueue 클래스를 작성하시오.



class BaseArray {

private:

int capacity; // 동적 할당된 메모리 용량

int \*mem;

protected:

BaseArray(int capacity=100) {

this->capacity = capacity; mem = new int [capacity];

}

~BaseArray() { delete [] mem; }

void put(int index, int val) { mem[index] = val; }

int get(int index) const{ return mem[index]; }

int getCapacity() { return capacity; }

};

|  |
| --- |
| [프로그램 소스]  #include <iostream> using namespace std;  class BaseArray { private:  int capacity; // 동적 할당된 메모리 용량  int \*mem; protected:  BaseArray(int capacity=100) {  this->capacity = capacity; mem = new int [capacity];  }  ~BaseArray() { delete [] mem; }  void put(int index, int val) { mem[index] = val; }  int get(int index) const{ return mem[index]; }  int getCapacity() { return capacity; } };  class MyQueue : public BaseArray{  int size=0; public:  MyQueue(int capacity) : BaseArray(capacity){}  void putQueue(int val){  put(size,val);  size++;  }  void deleteQueue(){  cout<<"큐의 원소를 순서대로 제거하여 출력한다>> ";  int size2=size;  for(int i=0 ; i<size2 ; i++){  cout<<get(i)<<" ";  size--;  }  cout<<endl;  }  int getSize(){  return size;  }  int getcapacity(){  return getCapacity();  } };  int main(){  int insert\_num;  int num;  cout<<"큐에 삽입할 정수개수를 입력하시오>>";  cin>>insert\_num;   MyQueue mq(100);  for(int i=0 ; i<insert\_num ; i++){  cout<<i+1<<" ) ";  cin>>num;  mq.putQueue(num);  }  cout<<"큐의 용량:"<<mq.getcapacity()<<", 큐의 크기 :"<<mq.getSize()<<endl;  mq.deleteQueue();  cout<<"큐의 현재 크기: "<<mq.getSize(); } |
| [실행결과]  텍스트이(가) 표시된 사진  자동 생성된 설명 |

1. 다음과 같은 클래스에 대하여 **상속을 적용하여** 제시된 결과처럼 실행될 수 있도록 프로그램을 작성하시오. 단, 잉크젯 프린터와 레이저 프린터 객체를 하나만 동적 생성하여 사용한다.

class InkJetPrinter {

int availableInk; //잉크 잔량

string model; //모델

string manufacturer; //제조사

int printedCount; //총 인쇄 매수

int availableCount; //인쇄 종이 잔량

public:

void printInkJet(int pages = 1); //pages 수만큼 용지 사용, 잉크잔량은 pages 수만큼 감소

void showStateInkJet(); //현재 상태 출력

};

class LaserPrinter {

int availableToner; //토너 잔량

string model; //모델

string manufacturer; //제조사

int printedCount; //인쇄매수

int availableCount; //인쇄 종이 잔량

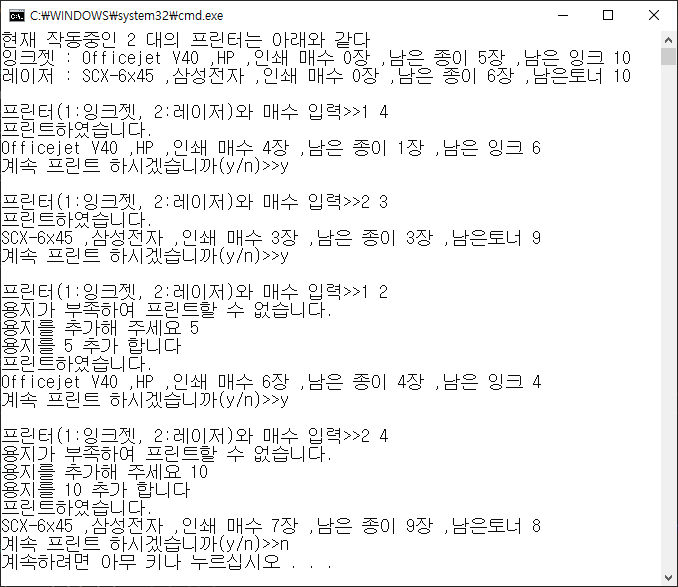
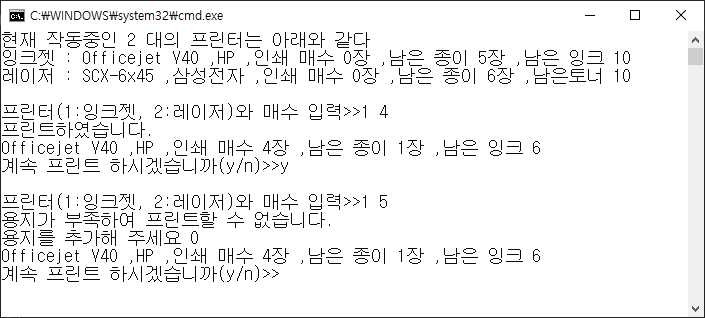
public:

void printLaser(int pages = 1); //pages 수만큼 용지 사용, 토너 잔량은 -1 감소

void showStateLaser(); //현재 상태 출력

};

**- 용지가 부족하면 입력한 값만큼 추가한 후 프린트 하고, 0을 입력하면 현재 프린터 상태만 출력한다**

int main() {

PrinterManager man;

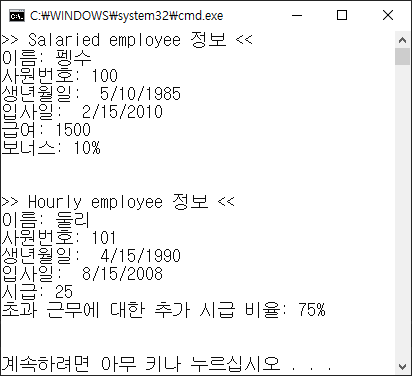
man.operater();

}

|  |
| --- |
| [프로그램 소스]  #include <iostream> #include <string> using namespace std;  class printer {  string model;  string manufacturer;  int printedCount;  int availableCount; public:  printer(string mod, string manu, int avc) {  model = mod;  manufacturer = manu;  availableCount = avc;  }  virtual void increaseAvailableCount(int pages)=0;  virtual int getAvail()=0; };  class InkJetPrinter :public printer{  int availableInk; //잉크 잔량  string model; //모델  string manufacturer; //제조사  int printedCount; //총 인쇄 매수  int availableCount; //인쇄 종이 잔량 public:  InkJetPrinter(string mod,string manu,int avc,int avi):printer(mod,manu,avc){  model = mod;  manufacturer = manu;  availableCount = avc;  availableInk=avi;  printedCount=0;  }  ~InkJetPrinter(){}  void printInkJet(int pages = 1){  printedCount+=pages;  availableInk-=pages;  availableCount-=pages;  } //pages 수만큼 용지 사용, 잉크잔량은 pages 수만큼 감소  void increaseAvailableCount(int pages){  availableCount+=pages;  }  void showStateInkJet(){  cout<<model<<" ,"<<manufacturer<<" ,"<<"인쇄 매수"<<printedCount<<"장 ,남은 종이"<<availableCount<<"장 ,남은 잉크"<<availableInk<<endl;  } //현재 상태 출력  int getAvail(){  return availableCount;  } };  class LaserPrinter:public printer {  int availableToner; //토너 잔량  string model; //모델  string manufacturer; //제조사  int printedCount; //인쇄매수  int availableCount; //인쇄 종이 잔량 public:  LaserPrinter(string mod,string manu,int avc,int avt):printer(mod,manu,avc){  model = mod;  manufacturer = manu;  availableCount = avc;  availableToner=avt;  printedCount=0;  }  ~LaserPrinter(){}  void printLaser(int pages = 1){  printedCount+=pages;  availableToner--;  availableCount-=pages;  }//pages 수만큼 용지 사용, 토너 잔량은 -1 감소  void increaseAvailableCount(int pages){  availableCount+=pages;  }  void showStateLaser(){  cout<<model<<" ,"<<manufacturer<<" ,"<<"인쇄 매수"<<printedCount<<"장 ,남은 종이"<<availableCount<<"장 ,남은 토너"<<availableToner<<endl;  } //현재 상태 출력  int getAvail(){  return availableCount;  } }; class PrinterManager{ public:  PrinterManager(){}  ~PrinterManager(){}  void operater(); }; void PrinterManager::operater() {  InkJetPrinter \*ink = new InkJetPrinter("Officejet Y40", "HP", 5, 10);  LaserPrinter \*Laser = new LaserPrinter("SCX-6x45", "삼성전자", 6, 10);  int print;  int pages;  string answer;  int increase\_pages;    cout << "현재 작동중인 2대의 프린터는 아래와 같다" << endl;  cout<<"잉크젯 : ";  ink->showStateInkJet();  cout<<"레이저 : ";  Laser->showStateLaser();   while(true) {  cout << "프린터(1:잉크젯, 2:레이저)와 매수 입력>>";  cin >> print >> pages;  if (print == 1) {  if(pages>(ink->getAvail())){  cout<<"용지가 부족하여 프린트 할수 없습니다."<<endl;  cout<<"용지를 추가해주세요 ";  cin>>increase\_pages;  if(increase\_pages==0){  ink->showStateInkJet();  }  else{  cout<<"용지를 "<<increase\_pages<<" 추가 합니다"<<endl;  ink->increaseAvailableCount(increase\_pages);  cout << "프린트하였습니다." << endl;  ink->printInkJet(pages);  ink->showStateInkJet();  }  }  else {  cout << "프린트하였습니다." << endl;  ink->printInkJet(pages);  ink->showStateInkJet();  }   } else if (print == 2) {  if(pages>(Laser->getAvail())){  cout<<"용지가 부족하여 프린트 할수 없습니다."<<endl;  cout<<"용지를 추가해주세요 ";  cin>>increase\_pages;  if(increase\_pages==0){  Laser->showStateLaser();  }  else {  cout << "용지를 " << increase\_pages << " 추가 합니다" << endl;  Laser->increaseAvailableCount(increase\_pages);  cout << "프린트하였습니다." << endl;  Laser->printLaser(pages);  Laser->showStateLaser();  }   }  else {  cout << "프린트하였습니다." << endl;  Laser->printLaser(pages);  Laser->showStateLaser();  }   } else {  cout << "다시 입력해주세요" << endl;  continue;  }  cout << "계속 프린트 하시겟습니까?(y/n)>>";  cin >> answer;  if (answer == "y")  continue;  else  break;  } }  int main() {  PrinterManager man;  man.operater(); } |
| [실행 결과]  텍스트이(가) 표시된 사진  자동 생성된 설명 |

1. 직원을 나타내는 Employee 클래스, 그리고 이를 기반으로 하는 월급을 받는 직원을 나타내는 Salaried 클래스, 시급을 받는 직원을 나타내는 Hourly 클래스를 작성하시오. 모든 Employee 클래스는 데이터 멤버로 이름(name), 직원 번호(employee number), 생일(birth date), 입사일(date hired)을 갖는다. 그리고 Salaried 클래스는 월급여(monthly salary)와 연간 보너스(annual bonus)(0~10%의 범위)를 추가로 갖고, Hourly 클래스는 시급(hourly wage)과 초과 근무에 대한 추가 시급 비율(overtime rate)(50~100%의 범위)을 추가로 갖는다.

int main(){

 cout << ">> Salaried employee 정보 <<" << endl;

Date birth1(5, 10, 1985); //생년월일

Date hired1(2, 15, 2010); //입사일

Salaried smith("펭수", 100, birth1, hired1, 1500.00, 10);

smith.print();

cout << endl << endl;

cout << ">> Hourly employee 정보 <<" << endl;

Date birth2(4, 15, 1990); //생년월일

Date hired2(8, 15, 2008); //입사일

Hourly howard("둘리", 101, birth2, hired2, 25.00, 75);

howard.print();

cout << endl << endl;

return 0;

}

|  |
| --- |
| [프로그램 소스]  #include <iostream> #include <string> using namespace std; class Date{ protected:  int month;  int year;  int date; public:  Date(int dt,int mon,int y):date(dt),month(mon),year(y){}  int getDate(){  return date;  }  int getMonth(){  return month;  }  int getYear(){  return year;  }  }; class Employee { protected:  string name;  int emplo\_num; public:  Employee(string name, int emplo\_num, Date birth, Date hired); };  Employee::Employee(string name, int emplo\_num, Date birth,Date hired){  this->name=name;  this->emplo\_num=emplo\_num; }  class Salaried :public Employee{  double mon\_salary;  int bonus;   int bir[3];  int hir[3]; public:  Salaried(string name, int emplo\_num, Date birth, Date hired, double mon\_salary, int bonus)  :Employee(name,emplo\_num,birth,hired){  this->bonus=bonus;  this->mon\_salary=mon\_salary;  bir[0]=birth.getDate();  bir[1]=birth.getMonth();  bir[2]=birth.getYear();  hir[0]=hired.getDate();  hir[1]=hired.getMonth();  hir[2]=hired.getYear();   }  void print(){  cout<<"이름: "<<name<<endl;  cout<<"사원번호: "<<emplo\_num<<endl;  cout<<"생년월일: "<<bir[0]<<"/"<<bir[1]<<"/"<<bir[2]<<endl;  cout<<"입사일: "<<hir[0]<<"/"<<hir[1]<<"/"<<hir[2]<<endl;  cout<<"급여: "<<mon\_salary<<endl;  if(bonus>=0 and bonus<=10)  cout<<"보너스: "<<bonus<<"%"<<endl;  else  cout<<"보너스의 범위는 0~10%입니다."<<endl;  } }; class Hourly : public Employee{  double hour\_wage;  int overtime\_rate;  int bir[3];  int hir[3]; public:  Hourly(string name,int emplo\_num,Date birth,Date hired,double hour\_wage,int overtime\_rate)  :Employee(name,emplo\_num,birth,hired)  {  this->hour\_wage=hour\_wage;  this->overtime\_rate=overtime\_rate;  bir[0]=birth.getDate();  bir[1]=birth.getMonth();  bir[2]=birth.getYear();  hir[0]=hired.getDate();  hir[1]=hired.getMonth();  hir[2]=hired.getYear();  }  void print(){  cout<<"이름: "<<name<<endl;  cout<<"사원번호: "<<emplo\_num<<endl;  cout<<"생년월일: "<<bir[0]<<"/"<<bir[1]<<"/"<<bir[2]<<endl;  cout<<"입사일: "<<hir[0]<<"/"<<hir[1]<<"/"<<hir[2]<<endl;  cout<<"시급: "<<hour\_wage<<endl;  if(overtime\_rate>=50 and overtime\_rate<=100)  cout<<"초과 근무에 대한 추가 시급 비율: "<<overtime\_rate<<"%"<<endl;  else  cout<<"초과 근무에 대한 추가 시급 비율은 50~100%입니다."<<endl;  } };   int main(){   cout << ">> Salaried employee 정보 <<" << endl;  Date birth1(5, 10, 1985); //생년월일  Date hired1(2, 15, 2010); //입사일  Salaried smith("펭수", 100, birth1, hired1, 1500.00, 10);  smith.print();  cout << endl << endl;   cout << ">> Hourly employee 정보 <<" << endl;  Date birth2(4, 15, 1990); //생년월일  Date hired2(8, 15, 2008); //입사일  Hourly howard("둘리", 101, birth2, hired2, 25.00, 75);  howard.print();  cout << endl << endl;  return 0; } |
| [실행 결과]  텍스트이(가) 표시된 사진  자동 생성된 설명 |