|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2018\_2\_C++ \_10 | 학번 : | 20157135 | 이름 : | 이상우 |

1. 상속 1

|  |
| --- |
| #include <iostream>  #include <string>  using namespace std;  class Point {  protected:  int x, y; //한 점 (x,y) 좌표값  public:  void set(int x, int y) { this->x = x; this->y = y; }  void showPoint() {  cout << "(" << x << "," << y << ")" << endl;  }  };  class ColorPoint : public Point {  string color;  public:  void setColor(string color) { this->color = color; }  void showColorPoint();  bool equals(ColorPoint p);  };  void ColorPoint::showColorPoint() {  cout << color << ":";  showPoint(); // Point 클래스의 showPoint() 호출  }  bool ColorPoint::equals(ColorPoint p) {  if(x == p.x && y == p.y && color == p.color) // ①  return true;  else  return false;  }  int main() {  Point p; // 기본 클래스의 객체 생성  p.set(2,3); // ②  p.x = 5; // ③  p.y = 5; // ④  p.showPoint();  ColorPoint cp; // 파생 클래스의 객체 생성  cp.x = 10; // ⑤  cp.y = 10; // ⑥  cp.set(3,4);  cp.setColor("Red");  ColorPoint cp2;  cp2.set(3,4);  cp2.setColor("Red");  cout << ((cp.equals(cp2))?"true":"false"); // ⑦  } |
| **[실행 결과]** |

* 상속 2 - private

|  |
| --- |
| #include <iostream>  using namespace std;  class Base {  int a;  protected:  void setA(int a) { this->a = a; }  public:  void showA() { cout << a; }  };  class Derived : private Base {  int b;  protected:  void setB(int b) { this->b = b; }  public:  void showB() { cout << b; }  };  int main() {  Derived x;  x.a = 5; // ①  x.setA(10); // ②  x.showA(); // ③  x.b = 10; // ④  x.setB(10); // ⑤  x.showB(); // ⑥  } |
| **[실행 결과]** |

* 상속 3 - protected

|  |
| --- |
| #include <iostream>  using namespace std;  class Base {  int a;  protected:  void setA(int a) { this->a = a; }  public:  void showA() { cout << a; }  };  class Derived : protected Base {  int b;  protected:  void setB(int b) { this->b = b; }  public:  void showB() { cout << b; }  };  int main() {  Derived x;  x.a = 5; // ①  x.setA(10); // ②  x.showA(); // ③  x.b = 10; // ④  x.setB(10); // ⑤  x.showB(); // ⑥  } |
| **[실행 결과]** |

* 내용 점검

1. 교재 8장 연습문제

|  |  |
| --- | --- |
| 번호 | 문제풀이 |
| 1 | 3.박 군의 집은 아버지가 주신 집이다. |
| 2 | 1.TV와 스마트TV |
| 3 | 3.protected 맴버는 클래스 내에서 보호받기 때문에 상속되지 않는다. |
| 4 | 4.D d; d.w=10; |
| 5 | x=a; |
| 6 | 5.setX(5); |
| 7 | 1. 2p = &b; . 3 q(B\*)&a; (2) q는 가 가르키는 것은 실직적으로는 a 객체 이기고 a 객체에는 y 가 없기 때문에 문제가 발생한다. |
| 8 | 1. bp=cp ; (2) int w , x , z (3) ap=dp; ap->x=10; (4) dp= (&d)ap; |
| 9 | (1)B b;  생성자 A  생성자 B  B b(10);  생성자 A  생성자 B10  B b(10,20);  생성자 A  생성자 B1020 |
| 10 | 1. class B 에서 에러, class A 의 디폴트 생성자가 X 2. B() :A(20) { cout << "생성자 B " << endl;; } 3. B(int x) :A(x+20) { cout << "생성자 B " << x << endl;; } |
| 11 | 3, 파생클래스-> 기본클래스 순으로 소멸된다. |
| 12 | 4. 다중 상속으로 인한 모호성은 실행 시간에 발견되므로 동적 ㅂ바인딩을 통한 가상 상속으로 해결 한다. |
| 13 | class Sstellite :public Rocket , public Computer{}; |
| 14 | 1. class Hipen : public Pen, public Eraser {}; 2. class Omnipen : public Pen, public Eraser, public Lock {}; |
| 15 | 4)  class Vehicle {  public:int power;  };  class Car :public Vehicle {  public:int color;  };  class Airplane {  public:int altitude;  };  class Flyingcar :public Car,public Airplane {  public:void go();  };  fcar.power=2000; // 오류   * Flyingcar 객체 가 생속 받는 Car,Airplane 클래스가 모두 Vehicle 를 상속을 받기 때문에   Vehicle 객체의 power 필드 접근시 Car,Airplane 중 어떤것으로 접근 할지 모르기 때문에  모호성 발생 한다. |
| 16 | ColorTV 와 internatTv 를 동시에 상속 받는 클래스 A 가 생성되고 A 객체로  TV 클래스의 screensize 접근시 모호성이 발생 한다. |
| 17 |  |

* 프로그램 과제

1. 교재 418p : 문제 1번

|  |
| --- |
| **[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:  namedcircle() :Circle(3){ name = "Noname"; }  namedcircle(string name) :Circle(3){ this->name = name; }  void show() {  cout << "반지름이 " << getradius() << "인 " << name << endl;  }  };  int main() {  namedcircle nd("waffle");  nd.show();  char c = getchar();  } |
| **[실행 결과]** |

1. 교재 418p : 문제 2번

|  |
| --- |
| **[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:  namedcircle() :Circle(3) { name = "Noname"; }  namedcircle(string name,int rd=0) :Circle(rd) { this->name = name; }  string getname() {  return name;  }  };  int main() {  int radius,index=0;  string name;  cout << "5 개의 정수 반지름과 원의 이름을 입력하세요." << endl; \  namedcircle nd[5];  for (int i = 0; i < 5; i++) {  cout <<i+1<<">> ";  cin >> radius >> name;  nd[i] = namedcircle(name,radius);  }  radius =nd[0].getarea();  for (int i = 0; i < 5; i++) {  if (nd[i].getarea() >= radius) {  radius = nd[i].getarea();  index = i;  }  }  cout << "가장 면적이 큰 피자는 " << nd[index].getname()<<" 입니다.";  char c = getchar();  c = getchar();  } |
| **[실행 결과]** |

1. 교재 419p : 문제 3번

|  |
| --- |
| **[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(int x, int y) {  this->x=x;  this->y = y;  }  };  class colorpoint : public point {  string color;  public:  colorpoint (int x , int y , string color) : point(x, y) {  this->color = color;  }  void setpoint(int x, int y) {  move(x, y);  }  void setcolor(string color) {  this->color = color;  }  void show() {  cout << color << "색으로 (" << getx() << "," << gety() << ")에 위치한 점입니다." << endl;  }  };  int main() {  colorpoint cp(5, 5, "RED");  cp.setpoint(10, 20);  cp.setcolor("BLUE");  cp.show();  char c = getchar();  } |
| **[실행 결과]** |

1. 교재 419p : 문제 4번

|  |
| --- |
| **[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(int x, int y) {  this->x = x;  this->y = y;  }  };  class colorpoint : public point {  string color;  public:  colorpoint() : point(0, 0) {  this->color = "BLACK";  }  colorpoint(int x, int y, string color="BLACK") : point(x, y) {  this->color = color;  }  void setpoint(int x, int y) {  move(x, y);  }  void setcolor(string color) {  this->color = color;  }  void show() {  cout << color << "색으로 (" << getx() << "," << gety() << ")에 위치한 점입니다." << endl;  }  };  int main() {  colorpoint zeropoint;  zeropoint.show();  colorpoint cp(5, 5, "RED");  cp.setpoint(10, 20);  cp.setcolor("BLUE");  cp.show();  char c = getchar();  } |
| **[실행 결과]** |

1. 교재 420p : 문제 6번

|  |
| --- |
| **[프로그램 소스]**  #include<iostream>  using namespace std;  class BaseArray {  private:  int capacity; //배열의 크기  int \*mem; //정수 배열을 만들기 위한 메모리의 포인터  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) { return mem[index]; }  int getCapacity() { return capacity; }  };  class MyStack : public BaseArray {  int count = 0;  public:  MyStack(int size) :BaseArray(size) {  count = 0;  }  void push(int n) {  put(count, n);  count++;  }  int capacity() {return getCapacity();}  int length() {return count;}  int pop() {  count--;  return get(count);  }  };  int main() {  MyStack mStack(100);  int n;  cout << "스택에 삽입할 5개의 정수를 입력하라 >> ";  for (int i = 0; i < 5; i++) {  cin >> n;  mStack.push(n);  }  cout << "스택용량 : " << mStack.capacity() << ", 스택크기 : " << mStack.length() << endl;  cout << "스택의 모든 원소를 팝하여 출력한다>> ";  while (mStack.length() != 0) {  cout << mStack.pop() << " ";  }  cout << endl << "스택의 현재 크기 : " << mStack.length() << endl;  char ch;  ch = getchar();  ch = getchar();  } |
| **[실행 결과]** |

1. 교재 422p : 문제 8번

|  |
| --- |
| **[프로그램 소스]**  #include<iostream>  using namespace std;  class BaseMemory {  char \*mem;  int size;  protected:  BaseMemory(int size) { mem = new char[size]; this->size = size; }  ~BaseMemory() { delete[] mem; }  void load(char \*mem, int address, int size) {  if (address + size > this->size) {  return;  }  for (int i = 0; i < size; i++)  this->mem[address + i] = mem[i];  }  public:  char read(int address) {  return mem[address];  }  };  class ROM : public BaseMemory {  public:  ROM(int size, char\* putchar, int putsize) : BaseMemory(size) {  load(putchar, 0, putsize);  }  };  class RAM : public BaseMemory {  public:  RAM(int size) : BaseMemory(size) {}  void write(int address, char data) {  load(&data, address, 1);  }  };  int main() {  char x[5] = { 'h', 'e', 'l', 'l', 'o' };  ROM biosROM(1024 \* 10, x, 5); //10KB의 ROM 메모리. 배열 x로 초기화됨  RAM mainMemory(1024 \* 1024); //1MB의 RAM 메모리  //0번지에서 4번지까지 biosROM에서 읽어 mainMemory에 복사  for (int i = 0; i < 5; i++) mainMemory.write(i, biosROM.read(i));  for (int i = 0; i < 5; i++) cout << mainMemory.read(i);  char c = getchar();  } |
| **[실행 결과]** |