|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2020\_2\_C++ \_04 | 학번 : | 20175105 | 이름 : | 곽영주 |

* **강의 내용**

1. 객체 포인터와 배열 활용 – 색의 3요소 다루기

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  class Color {  int red, green, blue;  string color;  public:  Color() { red = green = blue = 0; color = " "; }  Color(int r, int g, int b, string c) : red(r), green(g), blue(b), color(c) {}  ~Color();  void setColor(int r, int g, int b, string c);  void show();  };  void Color::setColor(int r, int g, int b, string c) {  color = c;  red = r;  green = g;  blue = b;  }  void Color::show() {  cout << "color = " << color << "\tRGB value >> R) " << red << " G) " << green << " B) " << blue << endl;  }  Color::~Color() {  cout << color << " 객체 소멸" << endl;  }  void pointer\_array() {  Color screenColor(255, 255, 255, "white");  Color \*p;  p = &screenColor;  p->show();  Color colors[3];  p = colors;  p[0].setColor(255, 0, 0, "red");  p[1].setColor(0, 255, 0, "green");  p[2].setColor(0, 0, 255, "blue");  for (int i = 0; i < 3; i++) {  p->show();  p++;  }  }  int main() {  pointer\_array();  cout << "program stop" << endl;  return 0;  } |
| **[실행 결과]** |

1. 동적 메모리 할당 활용 - 정수형 배열의 동적 할당 및 반환

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  using namespace std;  int main() {  int n;  cout << "입력할 실수의 개수는?";  cin >> n;  double\* p = new double[n];  if (!p) {  cout << "메모리를 할당할 수 없습니다.";  return 0;  }  for (int i = 0; i < n; i++) {  cout << i + 1 << "번째 실수: ";  cin >> p[i];  }  double sum = 0;  for (auto i = 0; i < n; i++) {  sum += p[i];  }  cout << "sum = " << sum << "\tave = " << sum / n << endl;  delete[] p;  } |
| **[실행 결과]** |

1. 객체 배열 동적 생성 활용

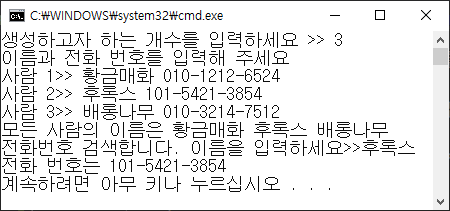
|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  using namespace std;  class Circle {  int radius;  public:  Circle() { radius - 1; }  Circle(int r) : radius(r) {}  ~Circle() {  cout << "반지름 " << radius << " 객체 소멸~~" << endl;  }  void setRadius(int r) {  radius = r;  }  double getArea();  int getRadius() const {  return radius;  }  };  double Circle::getArea() {  return 3.14 \* radius \* radius;  }  class CircleManage {  Circle \*p;  int size;  public:  CircleManage();  ~CircleManage();  void CircleWrite();  };  CircleManage::CircleManage() {  cout << "샹성하고자 하는 원의 개수 >> ";  cin >> size;  p = new Circle[size];  int radius;  for (int i = 0; i < size; i++) {  cout << "원" << i + 1 << ": ";  cin >> radius;  p[i].setRadius(radius);  }  }  void CircleManage::CircleWrite() {  int count = 0;  Circle \*pArray = p;  for (int i = 0; i < size; i++) {  cout << "반지름 = " << pArray->getRadius() << "\t넓이 = " << pArray->getArea() << endl;  if (pArray->getArea() >= 100 && pArray->getArea() <= 200)  count++;  pArray++;  }  cout << endl << "면적이 100에서 200 사이인 원의 개수는 " << count << endl << endl;  }  CircleManage::~CircleManage() {  delete[] p;  }  int main() {  CircleManage cm;  cm.CircleWrite();  return 0;  } |
| **[실행 결과]** |

* **프로그램 과제**

1. 입력 받은 크기만큼 객체 배열을 생성하고 이름과 전화번호는 입력 받아 초기화한다. 입력된 이름의 전화번호를 검색하여 출력하는 프로그램을 완성 하시오.

class Person {

string name;

 string tel;

public:

Person();

string getName() { return name; }

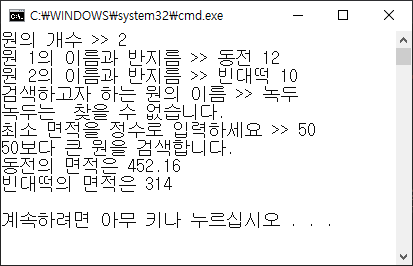
string getTel() { return tel; }

void set(string name, string tel);

};

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  class Person {  string name;  string tel;  public:  Person();  string getName() { return name; }  string getTel() { return tel; }  void set(string n, string t);  };  void Person::set(string n, string t) {  name = n;  tel = t;  }  Person::Person() {  }  int main() {  int n;  cout << "생성하고자 하는 개수를 입력하세요 >> ";  cin >> n;  Person\* p = new Person[n];  cout << "이름과 전화번호를 입력하세요." << endl;  string name, tel;  for (int i = 0; i < n; i++) {  cout << "사람 " << i + 1 << ">> ";  cin >> name >> tel;  p[i].set(name, tel);  }  cout << "모든 사람의 이름은 [";  for (int i = 0; i < n; i++) {  cout << " " << p[i].getName();  }  cout << " ] 전화번호 검색합니다." << endl;    string search\_name;  cout << "이름을 입력하세요 >> ";  cin >> search\_name;  for (int i = 0; i < n; i++) {  if (p[i].getName() == search\_name)  cout << "전화번호는 " << p[i].getTel() << endl;  }  delete[] p;  return 0;  } |
| **[실행 결과]** |

1. 원의 개수를 입력 받고, 개수만큼 원의 이름과 반지름을 입력 받는다. 다음과 같이 실행되도록 프로그램을 완성하시오.



class Circle {

int radius; // 원의 반지름 값

string name; // 원의 이름

public:

void setCircle(string name, int radius); // 이름과 반지름 설정

double getArea(); //넓이 반환

string getName(); //이름 반환

};

class CircleManager {

Circle \*p; // Circle 배열에 대한 포인터

int size; // 배열의 크기

public:

CircleManager(int size); // size 크기의 배열을 동적 생성. 사용자로부터 입력 완료

~CircleManager();

void searchByName(); // 사용자로부터 원의 이름을 입력 받아 면적 출력

void searchByArea(); // 사용자로부터 면적을 입력 받아 면적보다 큰 원의 이름 출력

};

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  class Circle {  int radius;  string name;  public:  void setCircle(string name, int radius);  double getArea();  string getName();  };  double Circle::getArea() {  return 3.14 \* radius \* radius;  }  string Circle::getName() {  return name;  }  void Circle::setCircle(string n, int r) {  name = n;  radius = r;  }  class CircleManager {  Circle\* p;  int size;  public:  CircleManager(int size);  ~CircleManager();  void searchByName();  void searchByArea();  };  CircleManager::CircleManager(int s) {  size = s;  p = new Circle[size];  string inputName; // 입력받는 원의 이름  int inputRadius; // 입력받는 반지름  for (int i = 0; i < size; i++) {  cout << "원 " << i + 1 << "의 이름과 반지름 >> ";  cin >> inputName >> inputRadius;  p[i].setCircle(inputName, inputRadius);  }  }  CircleManager::~CircleManager() {  delete[] p;  }  void CircleManager::searchByName() {  string searchName; // 검색하고자 하는 원의 이름  Circle \*tmp = p;  cout << "검색하고자 하는 원의 이름 >> ";  cin.ignore(1, '\n');  getline(cin, searchName);    int count = 0; // 못 찾은 개수  for (int i = 0; i < size; i++) {  if (tmp[i].getName() == searchName) { // 원의 이름이 같을 때  cout << searchName << " 면적 >> " << tmp[i].getArea() << endl;  }  else { // 다를 때  count++;  }  }  if (count == size ) { //하나도 못 찾은 경우  cout << searchName << "는 찾을 수 없습니다." << endl;  }  }  void CircleManager::searchByArea() {  int minArea; // 정수형 최소 면적 입력  cout << "최소 면적을 정수로 입력하세요 >> ";  cin >> minArea;  int count = 0; // 최소 면적보다 큰 개 없는 개수  cout << minArea << "보다 큰 원을 검색합니다." << endl;  for (int i = 0; i < size; i++) {  if (minArea < p[i].getArea()) { //있을 때  cout << p[i].getName() << "의 면적은 " << p[i].getArea() << endl;  }  else { // 없을 때  count++;  }  }  if (count == size) { //하나도 못 찾은 경우  cout << minArea << "보다 큰 원이 없습니다." << endl;  }  }  int main() {  int size;  cout << "원의 개수 >> ";  cin >> size;  CircleManager \*sm = new CircleManager(size);  sm->searchByName();  sm->searchByArea();  sm->~CircleManager();  return 0;  } |
| **[실행 결과]** |