|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2019\_1\_C++ \_04 | 학번 : | 20135151 | 이름 : | 이갑성 |

* **강의 내용**

|  |
| --- |
| 1. 객체 배열의 동적 생성과 반환 |
| #include <iostream>  using namespace std;  class Circle {  int radius;  public:  Circle();  Circle(int r);  ~Circle();  void setRadius(int r) { radius = r; }  double getArea();  };  double Circle::getArea() {  return 3.14\*radius\*radius;  }  Circle::Circle() {  radius=1;  cout<<”생성자 실행 radius = “<<radius<<endl;  }  Circle::Circle(int r) {  radius=r;  cout<<”생성자 실행 radius = “<<radius<<endl;  }  Circle::~Circle() {  radius=1;  cout<<”소멸자 실행 radius = “<<radius<<두이;  }  int main() {  Circle \*circleArray = new Circle[3]; // Circle 객체 배열 동적 생성, default 생성자 호출  circleArray[0].setRadius(10); circleArray[1].setRadius(20);  circleArray[2].setRadius(30);  for(int i=0; i<3; i++) // 배열의 각 원소 객체의 멤버 접근  cout << "Circle " << i << "의 면적은 " << circleArray[i].getArea() << endl;  Circle \*p= circleArray; // 포인터 p에 배열의 주소값 설정  for(int i=0; i<3; i++) { // 객체 포인터로 배열 접근  cout << "Circle " << i << "의 면적은 " << p->getArea() << endl;  p++; }  delete[] circleArray; //객체 배열 반환  } |
| **[실행결과]** |

|  |
| --- |
| 1. 객체 멤버 동적생성하기 |
| #include <iostream>  using namespace std;  class Dog {  string \*name;  int \*age;  public:  Dog(string n, int a){ //멤버 변수 동적 메모리 할당  name = new string(n); //name = new string{n}; // 동적 할당과 동시에 초기화 작업  age = new int(a); //age=new int{a}  }  ~Dog() {  delete name; //동적 할당 된 메모리 해제  delete age;  }  int getAge() { return \*age; }  void setAge(int a) { \*age = a; }  };  int main() {  Dog \*p=new Dog("강아지", 2);  cout << "강아지 나이 : " << p->getAge() << endl;  p->setAge(5);  cout << "강아지 나이 : " << p->getAge() << endl;  delete p;  } |
| **[실행결과]** |

|  |
| --- |
| 1. 문자열 입력 받고 회전시키기 |
| #include <iostream>  #include <string>  using namespace std;  int main() {  string s;  cout << "문자열을 입력하세요(한글 안됨) " << endl;  **getline(cin, s, '\n'); // 문자열 입력**  int len = **s.length()**; **// 문자열의 길이**  for(int i=0; i<len; i++) {  string first = **s.substr(0,1)**; **// 맨 앞의 문자 1개를 문자열로 분리**  string sub = **s.substr(1, len-1)**; **// 나머지 문자들을 문자열로 분리**  s = **sub + first**; **// 두 문자열을 연결하여 새로운 문자열로 만듦**  cout << s << endl;  }  } |
| **[실행결과]** |

|  |
| --- |
| 1. String 객체의 동적 생성 |
| #include <iostream>  #include <string>  using namespace std;  int main() {  string \*p = new string("C++"); // 스트링 객체 동적 생성  string \*names = new string[5]; //객체 배열 동적 생성  cout <<\*p<<endl; // "C++" 출력  p->append(" Great!!"); // p가 가리키는 스트링이 "C++ Great!!"이 됨  cout << \*p<<endl; // "C++ Great!!" 출력  for (int i = 0; i < 5; i++) {  cout << "이름 >> ";  getline(cin, names[i], '\n');  }  string latter = names[0];  for (int i = 1; i < 5; i++) {  if (latter < names[i]) { // 사전 순으로 latter 문자열이 앞에 온다면  latter = names[i]; // latter 문자열 변경  }  }  cout << "사전에서 가장 뒤에 나오는 문자열은 " << latter << endl;  delete p;  delete[] names;  } |
| **[실행결과]** |

|  |
| --- |
| 1. 문자열 find 및 replace |
| #include <iostream>  #include <string>  using namespace std;  int main() {  string s;  cout << "여러 줄의 문자열을 입력하세요. 입력의 끝은 &문자입니다." << endl;  getline(cin, s, '&'); // 문자열 입력  cin.ignore(); //& 뒤에 따라오는 <Enter>키를 제거하기 위한 코드  string f, r;  cout << endl << "find: ";  getline(cin, f, '\n'); // 검색할 문자열 입력  cout << "replace: ";  getline(cin, r, '\n'); // 대치할 문자열 입력  int startIndex = 0;  while (true) {  int fIndex = s.find(f, startIndex); // startIndex부터 문자열 f 검색  if (fIndex == -1)  break; // 문자열 s의 끝까지 변경하였음  // fIndex부터 문자열 f의 길이만큼 문자열 r로 변경  s.replace(fIndex, f.length(), r); //.replace(시작위치, 개수, 문자열);  startIndex = fIndex + r.length();  }  cout << s << endl;  } |
| **[실행결과]** |

* 내용 점검 – 4장 연습문제

|  |  |
| --- | --- |
| 번호 | 문제풀이 |
| 1 | (1) Rect r(2, 3);  (2) Rect \*p;  (3) cout << p->getArea(); |
| 2 | (1) q = new Rect(w, h);  (2) cout << "면적: " << q->getArea;  (3) delete q; |
| 3 | 1번(디폴트 생성자가 없기에) |
| 4 | 디폴트 생성자를 생성해주어야 한다.  class Rect {  int width, height;  public:  Rect() //디폴트 생성자  {  width = 0;  height = 0;  }  Rect(int w, int h)  {  width = w;  height = h;  }  int getWidth() { return width; }  int getHeigth() { return height; }  int getArea();  }; |
| 5 | for (int i = 0; i < 5; i++)  {  cout << "r[" << i << "]번째의 사각형 값: " << r[i].getArea() << endl;  } |
| 6 | 4번 |
| 7 | 4번 |
| 8 | 기본생성자  기본생성자  기본생성자  소멸자  소멸자  소멸자 |
| 9 | 1번 |
| 10 | delete [] p; |
| 11 | 3번 |
| 12 | 3번 |
| 13 | #include <iostream>  using namespace std;  class Location {  int width, height;  public:  Location() {  this->width = 0;  this->height = 0;  }  Location(int width, int height)  {  this->width;  this->height;  }  void show();  };  void Location::show()  {  cout << this->width << this->height << endl;  } |
| 14 | 동적 메모리 할당을 한 p가 있다고 가정했을 때, p가 다른 주소를 가르켰을 때 기존에 가르켰던 메모리는 반환도 안되있고, 사용도 되지 않고 있다. 이런 상황이 메모리 누수이다. |
| 15 | (1) delete [] p; //메모리 반환  (2) 누수X  (3) 누수X  (4) 누수X |
| 16 | 1번 |
| 17 | int n = stoi(s1);  int m = stoi(s2); |
| 18 | 3번 |
| 19 | string a("My name is Jane.");  char ch = a[2];  if (a == ("My name is John."))  {  cout << "same";  }  a = a + "~~";  a.replace(1, 1, "Y"); |

* 프로그램 과제

|  |
| --- |
| 1. 교재 211p 문제2번 |
| **[프로그램 소스]**  #include <iostream>  using namespace std;  int main()  {  int \*p = new int[5]; //정수형 5개 동적할당  double ave = 0.0;  cout << "정수 5개 입력 >> ";  for (int i = 0; i < 5; i++)  {  cin >> p[i];  }  for (int j = 0; j < 5; j++)  {  ave += p[j];  }  ave = ave / 5.0;    cout << "평균 " << ave;    delete[] p;  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 211p 문제4번 |
| **[프로그램 소스]**  #include <iostream>  using namespace std;  class Sample {  int \*p;  int size;  public:  Sample(int n)  {  size = n;  p = new int[n];  }  void read();  void write();  int big();  ~Sample();  };  void Sample::read() {  cout << "10개의 정수를 입력하세요." << endl;  for (int i = 0; i < 10; i++)  {  cin >> p[i];  }  }  void Sample::write() {  for (int i = 0; i < 10; i++)  {  cout << p[i] << " ";  }  cout << endl;  }  int Sample::big() {  int max = 0;  for (int i = 0; i < 10; i++)  {  if (max < p[i])  {  max = p[i];  }  }    return max;  }  Sample::~Sample() {  cout << "메모리를 반환합니다." << endl;  delete[] p;  }  int main()  {  Sample s(10);  s.read();  s.write();  cout << "가장 큰 수는 " << s.big() << endl;  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 213p 문제9번 |
| **[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 name, string tel);  };  Person::Person() {  name = "";  tel = "";  }  void Person::set(string name, string tel)  {  this->name = name;  this->tel = tel;  }  int main()  {  string name;  string tel;  cout << "이름과 전화번호를 입력해주세요" << endl;  Person per[3]; //per객체 배열 생성  for (int i = 0; i < 3; i++)  {  cout << "사람 " << i + 1 << ">> ";  cin >> name >> tel;  per[i].set(name, tel);  }  cout << "모든 사람의 이름은 ";  for (int i = 0; i < 3; i++)  {  cout << per[i].getName() << " ";  }  cout << endl;  cout << "전화번호 검색합니다. 이름을 입력하세요 >>";  cin >> name;  for (int i = 0; i < 3; i++)  {  if (name == per[i].getName())  {  cout << "전화번호는 " << per[i].getTel();  }  }  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 214p 문제10번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  class Person {  string name;  public:  Person() { name = " "; }  Person(string name)  {  this->name = name;  }  void setName(string name)  {  this->name = name;  }  string getName() {  return name;  }  };  class Family {  Person \*p;  int size;  public:  Family(string name, int size);  void show();  void setName(int index, string name);  ~Family();  };  Family::Family(string name, int size){  this->size = size;  p = new Person[size];  cout << name << "가족은 다음과 같이 " << size << "명 입니다." << endl;  }  void Family::setName(int index, string name)  {  p[index].setName(name);  }  void Family::show()  {  for (int i = 0; i < size; i++)  {  cout << p[i].getName() << " ";  }  }  Family::~Family()  {  delete[] p;  }  int main()  {  Family \*simpson = new Family("Simpson", 3); //3명으로 구성된 Simpson가족  simpson->setName(0, "Mr. Simpson");  simpson->setName(1, "Mrs. Simpson");  simpson->setName(2, "Bart Simpson");  simpson->show();  delete simpson;  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 216p 문제12번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  class Circle {  int radius; //원의 반지름  string name; //원의 이름  public:  void setCircle(string name, int radius); //이름과 반지름설정  double getArea() { return radius \* radius \* 3.14; };  string getName() { return name; };  };  void Circle::setCircle(string name, int radius)  {  this->name = name;  this->radius = radius;  }  class CircleManager {  Circle \*p; //Circle 배열에 대한 포인터  int size; //배열의 크기  public:  CircleManager(int size); //size 크기의 배열을 동적생성. 사용자로부터 입력완료  ~CircleManager();  void searchByName();  void searchByArea();  void show();  };  CircleManager::CircleManager(int size) {  this->size = size;  string name;  int rad;  p = new Circle[size]; //동적할당 Circle배열을 생성  for (int i = 0; i < size; i++)  {  cout << "원 " << i + 1 << "의 이름과 반지름 >>";  cin >> name >> rad;  p[i].setCircle(name, rad);  }  for (int i = 0; i < size; i++)  {  cout << "원 " << i + 1 << "의 이름: " << p[i].getName() << " 면적: " << p[i].getArea() << endl;  }  }  void CircleManager::searchByName() {  string findName; //검색하고자 하는 이름  cout << "검색하고자 하는 원의 이름 >>";  cin >> findName;  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*왜 for문에 접근을 못하는 건가요?\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  for (int i = 0; i < size; i++)  {  if (findName == p[i].getName())  {  cout << p[i].getName() << "의 면적은 " << p[i].getArea() << endl;  }  }  }  void CircleManager::searchByArea()  {  int min; //최소면적  cout << "최소 면적을 정수로 입력하세요 >>";  cin >> min;  cout << min << "보다 큰 원을 검색합니다." << endl;  for (int i = 0; i < size; i++)  {  if (p[i].getArea() > min)  {  cout << p[i].getName() << "의 면적은 " << p[i].getArea() << ", ";  }  }  }  void CircleManager::show() {  for (int i = 0; i < size; i++)  {  cout << p[i].getName() << "의 면적은 " << p[i].getArea() << endl;  }  }  CircleManager::~CircleManager() {  delete[] p;  }  int main()  {  int cnt;  cout << "원의 개수 >>";  cin >> cnt;  CircleManager \*cir = new CircleManager(cnt);  cir->searchByName();  cir->searchByArea();  delete cir;  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 218p 문제14번 : 선수 인원수는 프로그램 실행 시 입력 받도록 한다. |
| **[프로그램 소스]**  #include <iostream>  #include <string>  #include <cstdlib>  #include <ctime>  using namespace std;  class Player {  string playerName; //선수이름  public:  Player() {}  void setName(string playerName);  string getName();  };  void Player::setName(string playerName)  {  this->playerName = playerName;  }  string Player::getName() {  return playerName;  }  class GamblingGame {  Player \*per; //플레이어 포인터 변수 per 생성  int playerCnt; //플레이어의 수  public:  GamblingGame(int playerCnt);  void nameSetting();  void play();  ~GamblingGame();  };  GamblingGame::GamblingGame(int playerCnt)  {  this->playerCnt = playerCnt;  per = new Player[playerCnt]; //Player의 객체 배열 생성  }  void GamblingGame::nameSetting() { //이름저장하는 함수  string name;  for (int i = 0; i < playerCnt; i++)  {  cout << i + 1 << "번째 선수 이름 >> ";  cin >> name;  per[i].setName(name);  }    }  void GamblingGame::play(){  srand((unsigned int)time(NULL));  int arr[3]; //랜덤 값을 담을 배열  int input[3]; //게임하는사람이 넣을 값을 담을 배열  int personIndex = 0;  int gameResult = 0; //게임결과 1이면 정답자 나옴    while (1)  {  cout << per[personIndex].getName() << ": <Enter>" << endl;  for (int i = 0; i < 3; i++)  {  arr[i] = rand() % 3; // 0부터 2까지 난수 데이터 저장  }    for (int i = 0; i < 3; i++) //숫자 넣기  {  cin >> input[i];  }  for (int i = 0; i < 3; i++)  {  if (arr[i] != input[i])  {  gameResult = 0;  break;  }  else  {  gameResult = 1;  }  }  if (gameResult == 0)  {  cout << "아쉽군요!" << endl;  personIndex++;  if (personIndex == playerCnt) //사람인덱스를 첫번째 사람으로 변경해주는 작업  {  personIndex = 0;  }  }  else  {  cout << per[personIndex].getName() << "님이 맞추셨습니다. 축하드립니다!" << endl;  break;  }      }  }  GamblingGame::~GamblingGame() {  delete[] per;  }  int main()  {  int playerCnt; //선수 인원  cout << "\*\*\*\*\* 갬블링 게임을 시작합니다. \*\*\*\*\* \n";  cout << "선수 인원을 입력하세요 >>";  cin >> playerCnt;  GamblingGame \*gg = new GamblingGame(playerCnt);  gg->nameSetting();  gg->play();  } |
| **[실행결과]** |