|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2019\_1\_C++ \_04 | 학번 : | 20145128 | 이름 : | 박 인 근 |

* **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};  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;  } |
| **[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);  startIndex = fIndex + r.length();  }  cout << s << endl;  } |
| **[실행결과]** |

* 내용 점검 – 4장 연습문제

|  |  |
| --- | --- |
| 번호 | 문제풀이 |
| 1 | (1) Rect \*p;  (2) p = &r  (3) cout << "폭 :" << p->getWidth() << "높이 :" << p->getHeight() << endl; |
| 2 | (1) q = new Rect(w, h);  (2) cout << "면적 : " << q->getArea() << endl;  (3) delete q; |
| 3 | 1 |
| 4 | Rect 클래스의 기본 생성자를 생성합니다. |
| 5 | int a = 0;  Rect r[5] = { Rect(), Rect(2, 3), Rect(3,4), Rect(4,5), Rect(5,6) };  for (int i = 0; i < sizeof(r) / sizeof(Rect); i++) {  a += r[i].getArea();  }  cout << "배열 r의 사각형 면적의 합 : " << a << endl;  system("pause"); |
| 6 | 4 |
| 7 | 4 |
| 8 |  |
| 9 | 1 |
| 10 | delete[] p |
| 11 | 3 |
| 12 | 3 |
| 13 | Location() { this->width = this->height = 0; }  Location(int w, int h){ this->width = w; this->height = h; } |
| 14 | 메모리를 할당 후, 해제하지 않아서 시스템의 메모리를 고갈시키는 오류 |
| 15 | 1. delete[] p (2) delete p (4) 반복문 if 문 밑에 delete p; 써줘야합니다. |
| 16 | 1 |
| 17 | int n = stoi(s1);  int m = stoi(s2); |
| 18 | 3 |
| 19 | string a("My name is Jane.");  char ch = a[2];  if (a == "My name is john.") cout << "same";  else cout << "not same" << endl;  a += "~~";  a[1] = 'Y';  cout << a << endl |

* 프로그램 과제

|  |
| --- |
| 1. 교재 211p 문제2번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  int main() {  int \*a = new int[5];  double sum = 0;  double avg = 0;  cout << "정수 5개 입력>>";  for (int i = 0; i < 5; i++) {  cin >> a[i];  }  for (int i = 0; i < 5; i++) {  sum += a[i];  }  avg = sum / 5;  cout << "평균 : " << avg << endl;  delete[] a;  system("pause");  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 211p 문제4번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  class Sample {  int \*p;  int size;  int max;  public:  Sample(int n) {  size = n; p = new int[n];  }  void read() {  for (int i = 0; i < size; i++) {  cin >> p[i];  }  };  void write() {  for (int i = 0; i < size; i++) {  cout << ' ' << p[i];  }  };  int big() {  max = p[0];  for (int i = 0; i < size; i++) {  if (max < p[i]) max = p[i];  }  return max;  };  ~Sample() { delete p; };  };  int main() {  Sample s(10);  s.read();  s.write();  cout << "\n가장 큰 수는 " << s.big() << endl;  system("pause");  } |
| **[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);  };  void Person::set(string name, string tel)  {  this->name = name; this->tel = tel;  }  int main() {  Person info[3];  string name, tel;  cout << "이름과 전화번호를 입력해 주세요" << endl;    for (int i = 0; i < 3; i++) {  cout << "사람 " << i+1 << ">> ";  cin >> name >> tel;  info[i].set(name, tel);  }  cout << "모든 사람의 이름은 ";  for (int i = 0; i < 3; i++) {  cout << info[i].getName() << " ";  }  cout << "\n전화번호 검색합니다. 이름을 입력하세요>>";  cin >> name;  for (int i = 0; i < 3; i++) {  if (info[i].getName() == name) {  cout << "전화번호는 " << info[i].getTel();  }  }  getchar();  getchar();  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 214p 문제10번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  class Person {  string name;  public:  Person() {}  Person(string name) { this->name = name; }  string getName() { return name; }  void setName(string name) { this->name = name; }  };  class Family {  Person \*p;  int size;  string name; // main 에서 생성자에 name 도 받기 때문에 선언해줍니다.  public:  Family(string name, int size) {  p = new Person[size]; // size 개수만큼 Person 배열 동적 생성합니다.  this->name = name;  this->size = size;  }  void show() {  cout << name << "가족은 다음과 같이 " << size << "명 입니다." << endl;  for (int i = 0; i < size; i++)  cout << p[i].getName() << " ";  }  void setName(int size, string name) {  p[size].setName(name); // main에서 받은 매개변수 값을 각 Person 배열에 해당 정보를 저장해줍니다.  }  ~Family() { delete[] p; };  };  int main() {  Family \*simpson = new Family("Simpson", 3);  simpson->setName(0, "Mr.Simpson");  simpson->setName(1, "Mrs.Simpson");  simpson->setName(2, "Bart Simpson");  simpson->show();  delete simpson;  system("pause");  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 216p 문제12번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  class Circle {  int radius;  string name;  public:  void setCircle(string name, int radius) {  this->name = name; this->radius = radius;  }  double getArea() { return radius \* radius \* 3.14; }  string getName() { return name; }  void setName(string name) { this->name = name; }  };  class CircleManager {  Circle \*p;  int size;  string name;  int radius;  int rsize; //최소 면적용  public:  CircleManager(int size) {  p = new Circle[size];  this->size = size;  }  ~CircleManager() { delete[] p; }  void setName(int size, string name) {  p[size].setName(name);  }  void searchByName(string name) {  for (int i = 0; i < size; i++) {  if (p[i].getName() == name)  cout << p[i].getArea() << endl;  }  }  void searchByArea(int rsize) {  for (int i = 0; i < size; i++) {  if (p[i].getArea() > rsize)  cout << p[i].getName() << "의 면적은 " << p[i].getArea() << ",";  }  cout << endl;  }  void show() {  cout << "원의 개수 >> ";  cin >> size;  for (int i = 0; i < size; i++) {  cout << "원 " << i + 1 << "의 이름과 반지름 >> ";  cin >> name >> radius;  p[i].setCircle(name, radius);  }  cout << "검색하고자 하는 원의 이름 >>";  cin >> name;  cout << "도넛의 면적은 ";  searchByName(name);  cout << "최소 면적을 정수로 입력하세요 >> ";  cin >> rsize;  cout << rsize << "보다 큰 원을 검색합니다." << endl;  searchByArea(rsize);  }  };  int main() {  CircleManager \*circle = new CircleManager(4);  circle->show();  delete circle;  system("pause");  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 218p 문제14번 : 선수 인원수는 프로그램 실행 시 입력 받도록 한다. |
| **[프로그램 소스]**  #include <iostream>  #include <string>  #include <ctime>  #include <cstdlib>  using namespace std;  class Player {  string name[3];  int size;  public:  Player() {};    string getName() {  size = 0;  if (size == 0) return name[0];  else if (size == 1) return name[1];  else return name[2];  }  void setName(string name2) {  for (int i = 0; i < 3; i++)  name[i] = name2;  }  };  class GamblingGame {  Player \*p;  int size;  string name,enter;  int n[3]; // 3개의 랜덤수  public:  GamblingGame(int size) {  p = new Player[size];  this->size = size;  srand((unsigned)time(0));  }  ~GamblingGame() { delete[] p; }  void Gambling() {  cout << "\*\*\*\*\* 갬블링 게임을 시작합니다. \*\*\*\*\*" << endl;  cout << "\*\*\*\*\* 인원수를 입력하세요 \*\*\*\*\*" << endl;  cin >> size;  cin.ignore();  for (int i = 0; i < size; i++) {  cout << i+1 << "번째 선수 이름>>";  getline(cin, name);  p[i].setName(name);  }  while(1){  for (int j = 0; j < size; j++) {  for (int i = 0; i < 3; i++)  n[i] = rand() % 3; // 0~2까지  cout << p[j].getName() << ":<Enter>" << endl;  getline(cin, enter);  cout << " " << n[0] << " " << n[1] << " " << n[2];  if (n[0] == n[1] && n[1] == n[2]) {  cout << " " << p[j].getName() << "님 승리!!" << endl;  return;  break;  }  else cout << " 아쉽군요!" << endl;    }    }  }  };  int main() {  GamblingGame \*p = new GamblingGame(3); // 최대 3명  p->Gambling();  delete p;  system("pause");  } |
| **[실행결과]** |



