REPORT



|  |  |
| --- | --- |
| 과 목 : | 객체지향프로그래밍 |
| 제출일자 : | 2023.04.04 |
| 담당교수 : | 황성호 |
| 학 과 : | 멀티디자인학과 |
| 학 번 : | 201522405 |
| 이 름 : | 최준하 |

1.(예제 4-1) 객체 포인터 선언 및 활용

#include <iostream>

using namespace std;

class Circle {

int radius;

public :

Circle() { radius = 1; }

Circle(int r) { radius = r; }

double get\_area();

};

double Circle::get\_area() {

return 3.14 \* radius \* radius;

}

int main() {

Circle donut;

Circle pizza(30);

cout << donut.get\_area() << endl;

Circle\* p;

p = &donut;

cout << p->get\_area() << endl;

cout << (\*p).get\_area() << endl;

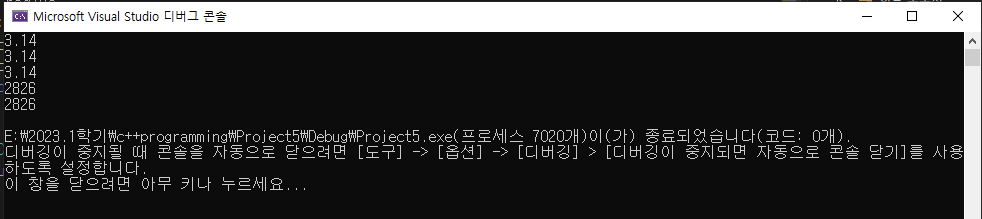
p = &pizza;

cout << p->get\_area() << endl;

cout << (\*p).get\_area() << endl;

}

🡪출력 결과



2.(예제 4-2) Circle 클래스의 배열 선언 및 활용.

#include <iostream>

using namespace std;

class \_circle {

int radius;

public:

\_circle() { radius = 1; }

\_circle(int r) { radius = r; }

void set\_radius(int r) { radius = r; }

double get\_area();

};

double \_circle::get\_area() {

return 3.14 \* radius \* radius;

}

int main() {

\_circle circle\_array[3];

circle\_array[0].set\_radius(10);

circle\_array[1].set\_radius(20);

circle\_array[2].set\_radius(30);

for (int i = 0; i < 3; i++)

{

cout << "circle" << i << "의 면적은" << circle\_array[i].get\_area() << endl;

}

\_circle\* p;

p = circle\_array;

for (int i = 0; i < 3; i++)

{

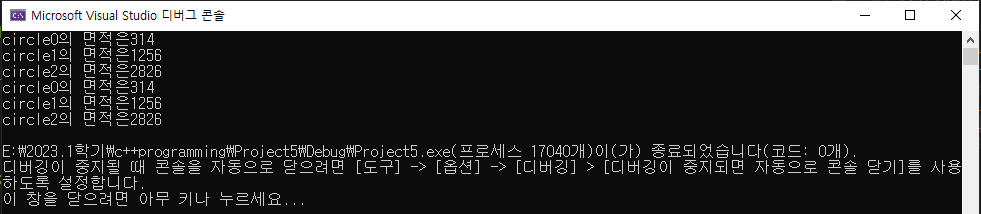
cout << "circle" << i << "의 면적은" << p->get\_area() << endl;

p++;

}

}

🡪출력 결과



3.(예제 4-3) 객체 배열 초기화.

#include <iostream>

using namespace std;

class \_circle {

int radius;

public:

\_circle() { radius = 1; }

\_circle(int r) { radius = r; }

void set\_radius(int r) { radius = r; }

double get\_area();

};

double \_circle::get\_area() {

return 3.14 \* radius \* radius;

}

int main() {

\_circle circle\_arr[3] = { \_circle(10),\_circle(20), \_circle() };

for (int i = 0; i <3; i++)

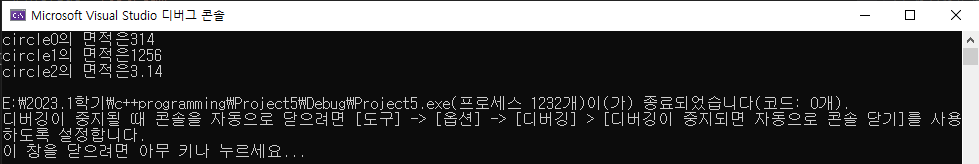
{

cout << "circle" << i << "의 면적은" << circle\_arr[i].get\_area() << endl;

}

}

🡪출력 결과



4.(예제 4-4) Circle 클래스의 2차원 배열 선언 및 활용.

#include <iostream>

using namespace std;

class \_circle {

int radius;

public:

\_circle() { radius = 1; }

\_circle(int r) { radius = r; }

void set\_radius(int r) { radius = r; }

double get\_area();

};

double \_circle::get\_area() {

return 3.14 \* radius \* radius;

}

int main() {

\_circle circles[2][3];

circles[0][0].set\_radius(1);

circles[0][1].set\_radius(2);

circles[0][2].set\_radius(3);

circles[1][0].set\_radius(4);

circles[1][1].set\_radius(5);

circles[1][2].set\_radius(6);

for (int i = 0; i < 2; i++)

{

for (int j = 0; j < 3; j++)

{

cout << "circle" << i << "," << j << "j의 면적은";

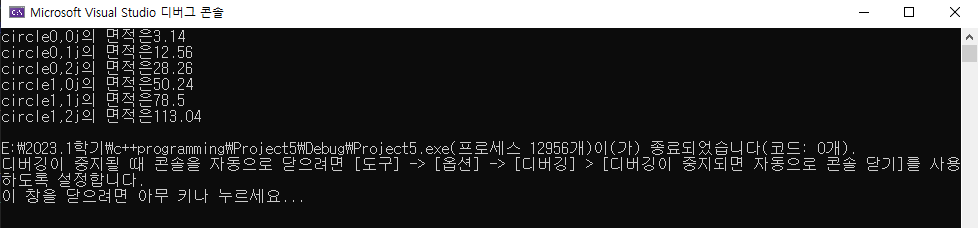
cout << circles[i][j].get\_area() << endl;

}

}

}

🡪출력 결과



5.(예제 4-5) 정수형 공간의 동적 할당 및 반환.

#include <iostream>

using namespace std;

int main() {

int\* p;

p = new int;

if (!p) {

cout << "메모리를 할당할 수 없습니다.";

return 0;

}

\*p = 5;

int n = \*p;

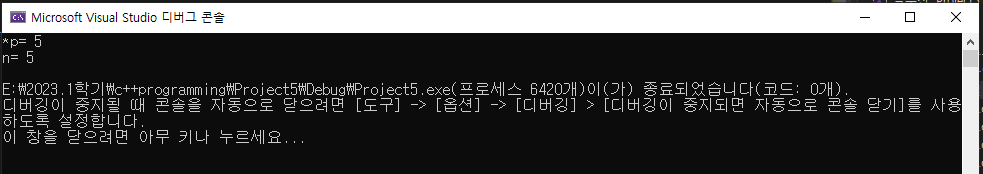
cout << "\*p= " << \*p << endl;

cout << "n= " << n << endl;

delete p;

}

🡪출력 결과



6.(예제 4-6) 정수형 배열의 동적 할당 및 반환.

#include <iostream>

using namespace std;

int main() {

cout << "입력할 정수의 개수는? ";

int n;

cin >> n;

if (n <= 0)

return 0;

int\* p = new int[n];

if (!p) {

cout << "할당 못함";

return 0;

}

for (int i = 0; i < n; i++)

{

cout << i + 1 << "번째 정수: ";

cin >> p[i];

}

int sum = 0;

for (int i = 0; i < n; i++)

{

sum += p[i];

}

cout << "평균 = " << sum / n << endl;

delete[]p;

}

🡪출력 결과

