객체지향프로그래밍 (202004100 장주훈, 202084023 이병길)

**7주차**

객체 포인터

c언어의 포인터와 동일하며, p = &donut 이면 dount에 대한 주소 값을 p에 저장

호출시에는 d = p->getArea();

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

Q1. 포인터가 무엇인지?

|  |
| --- |
| 자신이 가리킨 변수의 주소 값을 지니는 변수 |
| 변수가 가지고 있는 주소를 가리키는 변수  P=&a 일 경우 p에는 a의 주소 값 0x0000~~가 들어 감  But, \*p=20일 경우 a의 주소에 값 20을 넣어 줌. |

객체 배열 //매개 변수 없는 생성자 호출

배열 = 포인터이지만 상수가 들어가는 상수 포인터이다.

Q2. 객체 배열이 무엇인지?

|  |
| --- |
| 배열은 포인터긴 하지만 그 주소의 시작을 알려주고 주소에 해당하는 공간에 바로 값을 넣어주는 역할을 한다. 따라서, 객체 배열을 선언할 때에는 공간을 할당하고 호출시에는 매개 변수가 없는 생성자를 호출하게 된다. |

|  |
| --- |
| Main.cpp  #include<iostream>  #include"pro1.h"  using namespace std;  int main() {  Circle circleArray[3];  circleArray[0].setRadius(10);  circleArray[1].setRadius(20);  circleArray[2].setRadius(30);  for (int i = 0; i < 3; i++)  cout << "Circle " << i << "의 면적은 " << circleArray[i].getArea() << endl;  Circle \*p;  p = circleArray;  for (int i = 0; i < 3; i++) {  cout << "Circle " << i << "의 면적은 " << p->getArea() << endl;  p++;  }  } |
| Pro1.h  #ifndef pro1  #define pro1  class Circle {  int radius;  public:  Circle() { radius = 1; }  Circle(int r) { radius = r; }  void setRadius(int r) { radius = r; }  double getArea();  };  #endif |
| Project1.cpp  #include"pro1.h"  #include<iostream>  using namespace std;  double Circle::getArea() {  return 3.14 \* radius \* radius;  } |

P = circleArray;에서 & 주소값을 안 넣은 이유는 배열이기 때문에 그 자체로 주소 값을 나타내기 때문이다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

생성자가 선언되어 있으면 컴파일러가 자동으로 기본 생성자를 안 만들어 줌.

객체 배열 초기화 방법

Circle circleArray[3] = { Circle(10), Circle(20), Circle() };

이중 for문을 이용해 2차원 배열 선언

텍스트, 폰트, 스크린샷, 화이트이(가) 표시된 사진

자동 생성된 설명

동적 메모리 할당 및 반환

* 정적 할당: 변수 선언을 통해 필요한 메모리 할당
* 동적 할당: 필요한 양이 예측되지 않는 경우. 힙 메모리에서 할당

+ 힙은 운영체제가 프로세스의 실행을 시작 시킬 때 동적 할당 공간으로 준 메모리 공간

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

텍스트, 폰트, 스크린샷, 번호이(가) 표시된 사진

자동 생성된 설명

예제 4-5 직접 풀어보기

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  int\* p;  p = new int;  if (!p) {  cout << "메모리를 할당할 수 없습니다.";  return 0;  }  \*p = 5; //포인터가 가리키는 값(포인터 x)  int n = \*p; // p의 내용 (포인터 x)  cout << "\*p = " << \*p << '\n';  cout << "n = " << n << '\n';  delete p;  } |

Q3. 배열의 동적 할당 및 반환(ppt 18 페이지)

|  |
| --- |
| 포인터 변수 p가 배열로 선언되어 영역을 할당 받고, 이를 for문을 통해 0~4까지 값을 넣어줌.  이 후, 배열을 delete 연산자를 통해 배열의 영역인 메모리를 반환해준다. |
| 포인터 변수 p가 동적 메모리에 할당된 0~4번째 정수가 들어있는 배열을 가리킨다. 그 다음 포인터 변수 p를 통해 0~4까지의 값을 할당. 포인터 변수 p가 가르키는 배열의 메모리 할당을 해제한다. |

텍스트, 폰트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명차이점!



예제 4-6 중요!

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  cout << "입력할 정수의 개수는?"; //1번  int n;  cin >> n; // 정수의 개수 입력 2번  if (n <= 0) return 0; //3번  int\* p = new int[n]; // n 개의 정수 배열 동적 할당  if (!p) {  cout << "메모리를 할당할 수 없습니다.";  return 0;  }  for (int i = 0; i < n; i++) { //4번  cout << i + 1 << "번째 정수: "; // 프롬프트 출력  cin >> p[i]; // 키보드로부터 정수 입력  }  int sum = 0;  for (int i = 0; i < n; i++)  sum += p[i];  cout << "평균 = " << sum / n << endl;  delete[] p; // 배열 메모리 반환  } |

동기화 할당 메모리 초기화

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

4-10 예제 꼭 풀어보기!

this포인터: 객체 자신을 가리키는 포인터, 클래스 멤버 함수 내에서만 사용

\* 각 객체 속의 this는 다른 객체의 this와 다름