

2100998 이수찬

0512

4-3장 c++

- 과제1

- 메모리 누수란

- ◆ 더 이상 필요하지 않은 메모리를 해제하지 않고 계속 점유하고 있는 현상

- 메모리 누수가 왜 심각한 문제를 발생시키는가?

- ◆ 메모리가 계속해서 쌓이면 비정상 종료를 발생시킬 수 있다.
 - ◆ 나중에 원인을 찾기 어렵다.

- 메모리 누수를 방지하는 법

- ◆ 메모리 할당 받았으면 사용 후 해제해준다.

- 과제2

□ 아래 코드의 문제점을 설명하고 해결방법을 설명하시오.

```
// 예제4-8의 Circle 클래스 정의 추가
int main() {
    int radius;
    while (true) {
        cout << "반지름 입력(음수이면 종료)>> ";
        cin >> radius;
        if (radius < 0) break;
        Circle* p = new Circle(radius);
        cout << "원의 면적:" << p->getArea() << endl;
    }
}
```

- 현재 while 반복문이 계속해서 돌아가는 상황에 객체메모리가 생성된 후 계속해서 새롭게 덮어쓰워지고 있어 메모리가 낭비되는 상황이 발생한다. 이러한 상황에서 생성된 메모리들을 원의 면적을 구한 후 delete를 사용해 힙에 있는 메모리를 제거해준다.

```
int main() {
    int radius;
    while (true) {
        cout << "반지름 입력(음수이면 종료)>> ";//
```

```

        cin >> radius;
        if (radius < 0) break;
            Circle* p = new Circle(radius);
        cout << "원의 면적:" << p->getArea() << endl;
        delete p; //메모리 해제시켜서 누수 제거
    }
}

```

● 과제3

□ 아래 코드의 문제점을 설명하고 해결방법을 설명하시오.

```

// 예제4-9의 Circle 클래스 정의 추가
int main() {
    Circle* pArray = new Circle[3];
    for (int i = 0; i < 3; i++) {
        cout << pArray ->getArea() << '\n';
        pArray++;
    }

    delete [ ] pArray;
}

```

```

int main() {
    Circle* pArray = new Circle[3]; // 배열 동적 할당

    for (int i = 0; i < 3; i++) {
        cout << "원의 면적: " << pArray[i].getArea() << '\n' //주소변경 안되도록
        객체[i]로 변경
    }

    delete[] pArray; // 동적 메모리 해제
    return 0;
}

```

● 과제4

```

#include<iostream> //입출력 헤더파일
using namespace std; //이름공간선언

class Triangle { //클래스 선언
private: //접근지정자
    int width, height; //정수변수 선언
public: //접근지정자
    Triangle(); //기본생성자

```

```

        Triangle(int w, int h); //매개변수 가진 생성자
        float getArea(); //삼각형면적 매개함수
        ~Triangle(); //소멸자
};

Triangle::Triangle() { //생성자
    width = 0; //기본 생성자 밀변 선언
    height = 0; //높이 선언
}
Triangle::Triangle(int w, int h) { //매개변수 가진 생성자
    width = w; //밀변 초기화
    height = h; //높이 초기화
    cout << "밀변" << width << "높이" << height << "인 삼각형 생성" << endl;
}

float Triangle::getArea() { //면적 매개함수
    return width * height * 0.5; //면적값 리턴
}

Triangle::~~Triangle() { //소멸자
    cout << "밀변" << width << "높이" << height << "인 삼각형 소멸" << endl;
}

int main() {
    Triangle* A[3]; //포인터 배열 객체 선언
    A[0]=new Triangle(1, 1); // 배열 선언
    A[1]=new Triangle(2, 2); //배열 선언
    A[2]=new Triangle(4, 4); //배열 선언

    for (int i = 0; i < 3; i++) {
        cout << A[i]->getArea() << endl;
    }
    for (int i = 0; i < 3; i++) {
        delete A[i]; //각 메모리공간 해제
    }
}

```

● 과제5

```

#include <iostream> //입출력 헤더파일
using namespace std; //이름공간선언

class Sphere { //클래스 선언
private: //접근지정자
    int radius, num; //정수변수 선언
public: //접근지정자
    float getArea(); //구 부피 매개함수
    void setRadius(int r); //반지름설정
};

float Sphere::getArea() { //면적 매개함수
    return (4.0/3.0)*3.14*radius*radius*radius; //부피값 리턴
}

void Sphere::setRadius(int r) {
    radius = r; //생성자 반지름선언
}

```

```

}

int main() {
    cout << "생성하고자 하는 구의 갯수: ";
    int num, radius;
    cin >> num;
    Sphere *s = new Sphere[num]; // 동적할당 받기위한 포인터 변수 선언
    if (!s) { //메모리 오류 체크
        cout << "메모리부족";
        return -1;
    }
    for (int i = 1; i < num+1; i++) {
        cout << "구" << i << "의 반지름: ";
        cin >> radius;
        s[i - 1].setRadius(radius); // 반지름 설정
    }

    for (int i = 0; i < num; i++) {
        cout << s[i].getArea() << endl; //부피 구하기
    }

    delete []s; //각 메모리공간 해제
}

```

```

생성하고자 하는 구의 갯수: 3
구1의 반지름: 1
구2의 반지름: 2
구3의 반지름: 3
4.18667
33.4933
113.04

```