

0404 (3-2)

2100998 이수찬

- 과제1.

- C++에서 생성자를 통하여 멤버변수의 초기화를 하는 이유를 설명하라.
 - ◆
- 생성자 작성시 생성자 몸체에서 멤버를 초기화하는 것 보다 멤버 초기화 리스트에서 초기화하는 이유를 설명하라.
 - ◆
- 위임, 타겟 생성자로 나누어 작성하는 이유를 설명하라
 - ◆
- 함수 몸체(body)에서 생성자를 직접 호출하면 어떻게 되는지 조사해보라
 - ◆ ○

- 과제2

```
#include <iostream>
using namespace std;

class Triangle { // 클래스 생성
    double base, height, area; // 밑변, 높이, 면적

public:
    Triangle() : Triangle(1.0, 1.0) {} // 기본 생성자 생성 후 밑변과 높이 초기화

    Triangle(double b, double h) : base(b), height(h) { //타겟생성자 생성
        area = (1.0 / 2.0) * base * height; // 면적 계산
    }

    double getArea() { return area; } //멤버함수 선언
};

int main() {
    Triangle tri1; // 밑변, 높이 1로 초기화
    cout << "삼각형의 면적은 " << tri1.getArea() << endl; // 삼각형 면적 출력

    Triangle tri2(2, 4); // 밑변=2, 높이=4로 초기화
    cout << "삼각형의 면적은 " << tri2.getArea() << endl; // 삼각형 면적 출력 2

    return 0; // 함수 반환 후 종료
} // 메인함수 종료
```

Microsoft Visual Studio 디버그

삼각형의 면적은 0.5
삼각형의 면적은 4

C:\Users\leesoochan\OneDrive\바탕 화면\3학년
되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...|

- 과제3

```
#include <iostream>
#include <iomanip> // setprecision 사용을 위해 추가
using namespace std;

// Sphere 클래스 선언
class Sphere {
    double radius; // 반지름
    double volume; // 부피

public:
    Sphere() : Sphere(1.0) { } // 기본 생성자 생성

    Sphere(double r) : radius(r) { // 타겟생성자 생성
        volume = (4.0 / 3.0) * 3.14 * radius * radius * radius; // 부피 계산
    }

    double getVolume() { // 부피를 반환하는 멤버 함수
        return volume; // 볼륨 리턴
    }
};

int main() { // 메인함수 시작
    Sphere sph1; // 반지름 = 1로 초기화
    cout << fixed << setprecision(2); // 소수점 둘째 자리까지 고정
    cout << "구의 부피는 " << sph1.getVolume() << endl; // 구의 부피 출력

    Sphere sph2(3); // 반지름 = 3로 초기화
    cout << "구의 부피는 " << sph2.getVolume() << endl; // 구의 부피 출력2

    return 0; // 프로그램 종료
}
```

구의 부피는 4.19
구의 부피는 113.04

C:\Users\leesoochan\OneDrive\바탕 화면\3학
도었습니다(코드 : 0개).

- 과제4

```
#include <iostream>
#include <iomanip> // setprecision 사용을 위해 추가
using namespace std;

// Rectangle 클래스 선언
class Rectangle {
    int x, y;           // 좌측 상단 좌표
    int width, height; // 가로(width), 세로(height)

public:
    Rectangle() : Rectangle(1, 1, 1, 1) { } // 기본 생성자 (모든 값을 1로 초기화)

    Rectangle(int x, int y) : Rectangle(x, y, 1, 1) { } // (x, y)만 받는 생성자
    (width, height = 1)

    Rectangle(int x, int y, int width, int height) // 모든 멤버를 받는 타겟 생성자
        : x(x), y(y), width(width), height(height) { } // 멤버 초기화

    bool isSquare() { // 정사각형인지 검사하는 멤버 함수
        return width == height; // 가로와 세로가 같으면 true 반환
    }

    int getArea() { // 면적 반환하는 멤버 함수
        return width * height; // 가로 * 세로
    }

    int getPerimeter() { // 둘레 길이 반환하는 멤버 함수
        return 2 * (width + height); // 2 * (가로 + 세로)
    }

    void getRightBottom( int& rx, int& ry) { // 우측 하단 좌표를 계산하는 멤버 함수
        rx = x + width; // 오른쪽 끝 좌표 = x + 가로 길이
        ry = y - height; // 아래쪽 끝 좌표 = y - 세로 길이
    }
};

int main() { // 메인 함수 시작
    Rectangle rect1; // (1,1,1,1)로 초기화
    Rectangle rect2(3, 5); // (3,5,1,1)로 초기화
    Rectangle rect3(3, 5, 2, 4); // (3,5,2,4)로 초기화
```

```
cout << "rect1의 면적은 " << rect1.getArea() << endl; // rect1의 면적 출력  
cout << "rect2의 둘레길이는 " << rect2.getPerimeter() << endl; // rect2의  
둘레길이 출력  
  
int rightX, rightY; // 우측 하단 좌표 저장할 변수 선언  
rect3.getRightBottom(rightX, rightY); // rect3의 우측 하단 좌표 계산  
cout << "rect3의 우측하단의 좌표는 (" << rightX << ", " << rightY << ")" << endl;  
// 우측 하단 좌표 출력  
  
return 0; // 프로그램 종료  
}
```

```
rect1의 면적은 1  
rect2의 둘레길이는 4  
rect3의 우측하단의 좌표는 (5, 1)
```

```
C:\Users\leesoochan\OneDrive\바탕 화면\3학년  
되었습니다(코드 : 0개).
```