

객체지향프로그래밍 정리노트(Week3)

202004029 김정호, 202004040 노성민

Q1. 예제 3-1 Circle 클래스의 객체 생성 및 활용

김정호: 우선 Circle 클래스의 필요한 변수나 함수를 생각해보자. 문제가 원의 면적을 구하는 문제이니까 우선 원의 반지름을 변수로 하고 면적을 구하는 함수를 만들자.

노성민: 음 그러면 멤버 변수와 함수에 대한 접근 지정자를 정해야 하는데 디폴트가 private인데 보통 멤버 변수정도만 private으로 하는 걸로 알고 있어.

김정호: 그러면 변수는 디폴트 값으로 하고 함수를 public을 이용하자.

노성민: 그럼 우선 class 먼저 만들어 보자.

코드

```
#include <iostream>
using namespace std;
class Circle {
int radius;
public:
double getArea();
};
double Circle::getArea() {
return 3.14*radius*radius;
}
```

김정호: 그럼 이제 메인 함수에서 객체를 만들어야 하고 그 객체의 반지름과 함수를 이용해서 면적을 출력하자.

코드

```
#include <iostream>
using namespace std;
class Circle {
int radius;
public:
double getArea();
};
double Circle::getArea() {
return 3.14*radius*radius;
}

int main() {
Circle donut;
donut.radius = 1;
double area = donut.getArea();
cout << "donut 면적은 " << area << endl;
```

```

Circle pizza;
pizza.radius = 30;
area = pizza.getArea();
cout << "pizza 면적은 " << area << endl;
}

```

Q2. 3-2(실습) - Rectangle 클래스 만들기

다음 main() 함수가 잘 작동하도록 너비(width)와 높이(height)를 가지고 면적 계산 기능을 가진 Rectangle 클래스를 작성하고 전체 프로그램을 완성하라.

```

int main() {
    Rectangle rect;
    rect.width = 3;
    rect.height = 5;
    cout << "사각형의 면적은 " << rect.getArea() << endl;
}

```

결과: 사각형의 면적은 15

김정호: 우선 main 함수에 Rectangle 클래스에 객체로 rect이 선언됐고 rect의 width와 height의 값을 초기화 해줘서 getArea()라는 함수의 값을 출력했을 때 15라는 정수형 값이 나왔으니 이에 맞게 클래스를 작성하면 될거 같아.

노성민: 한번 작성해 보자.

코드

```

#include<iostream>
using namespace std;

class Rectangle {
public:
    int width,height;
    int getArea();
};

int Rectangle::getArea() {
    return width * height;
}

int main() {
    Rectangle rect;
    rect.width = 3;
    rect.height = 5;
    cout << "사각형의 면적은 " << rect.getArea() << endl;
}

```

Q3. 예제 3-3을 수정하여 객체 초기화를 전담하는 타겟 생성자와 타겟 생성자에 개

제 초기화를 위임하는 위임 생성자로 재작성하라.

코드

```
#include <iostream>
using namespace std;
class Circle {
public:
    int radius;
    Circle(); // 매개 변수 없는 생성자
    Circle(int r); // 매개 변수 있는 생성자
    double getArea();
};
Circle::Circle() {
    radius = 1;
    cout << "반지름 " << radius << " 원 생성" << endl;
}
Circle::Circle(int r) {
    radius = r;
    cout << "반지름 " << radius << " 원 생성" << endl;
}
double Circle::getArea() {
    return 3.14*radius*radius;
}
int main() {
    Circle donut; // 매개 변수 없는 생성자 호출
    double area = donut.getArea();
    cout << "donut 면적은 " << area << endl;
    Circle pizza(30); // 매개 변수 있는 생성자 호출
    area = pizza.getArea();
    cout << "pizza 면적은 " << area << endl;
    Circle(30); 자동 호출 }
```

김정호: 예제를 보면 Circle::Circle(int r) 매개 변수가 있는 생성자를 타겟 생성자로 설정해서 Circle::Circle() 매개 변수가 없는 생성자를 타겟 생성자로 설정하면 될 거 같아.

따라서 빨간색 부분은 Circle::Circle() : Circle(1) { }으로 수정하면 될 거 같아.

노성민: 그런 것 같아.

Q4. ppt 29페이지 왜 오류가 나는 이유?

```
class Circle {
public:
    int radius;
    double getArea();
    Circle();
    Circle(int r);
};
Circle::Circle() {
```

```

}
Circle::Circle(int r) {
    radius = r;
}
int main() {
    Circle pizza(30);
    Circle donut;
}

```

김정호: 내가 알기로는 생성자가 하나라도 선언이 되면 기본 생성자는 자동으로 생성되지 않기 때문에 우리가 기본 생성자를 적어 줘야 해.

노성민: 그게 맞는 거 같아.

Q5. 3-6(실습) - Rectangle 클래스 만들기

다음 main() 함수가 잘 작동하도록 Rectangle 클래스를 작성하고 프로그램을 완성하라.

Rectangle 클래스는 width와 height의 두 멤버 변수와 3 개의 생성자, 그리고 isSquare() 함수를 가진다.

```

int main() {
    Rectangle rect1;
    Rectangle rect2(3, 5);
    Rectangle rect3(3);
    if(rect1.isSquare()) cout << "rect1은 정사각형이다." << endl;
    if(rect2.isSquare()) cout << "rect2는 정사각형이다." << endl;
    if(rect3.isSquare()) cout << "rect3는 정사각형이다." << endl;
}

```

결과:

rect1은 정사각형이다.

rect3는 정사각형이다.

김정호: 우선 메인 함수를 보면 Rectangle이라는 클래스에 대고 rect 1, 2, 3 객체 3개가 만들어진 걸 알 수 있어. 그리고 생성자를 3개를 만들어야 해. 매개 변수가 없는 것, 하나 있는 것, 두 개 있는 것으로 하면 돼. 그 아래 줄을 보면 if 문 안에 객체명.isSquire()를 보면 우선 isSquare 라는 함수가 클래스 안에 있고 bool형이라는 것을 알 수 있어. 이제 한번 클래스 안에는 뭐가 필요할지 생각해보자.

노성민: 결과를 보면 알다시피 정사각형임을 판별하는 것이 핵심이기 때문에 가로,세로 길이를 알아야 할 것 같아. 그리고 rect 1은 기본 생성자인데 정사각형인 것을 보면 초기화한 값이 있는 것 같고, 매개 변수가 하나인 생성자는 가로, 세로 길이가 같은 것으로 보여. 그러면 한번 코드를 작성해보자.

코드

```

#include<iostream>
using namespace std;

class Rectangle{
    int width, height;
public:

```

```

        bool isSquare();
        Rectangle();
        Rectangle(int x);
        Rectangle(int x, int y);
};

```

김정호: 우선 여기까지는 변수, 함수, 생성자를 선언했어. 이제 각 함수, 생성자에 맞게 작성해보자. 그리고 생성자는 작성할 때 위에서 배웠던 타겟 생성자와 위임 생성자를 이용해서 작성해보자.

코드

```

Rectangle::Rectangle() : Rectangle(1){}

```

```

Rectangle::Rectangle(int x)
    : width(x), height(x){}

```

```

Rectangle::Rectangle(int x, int y) :
    width(x), height(y) {}

```

```

bool Rectangle::isSquare() {
    if (width == height)
        return true;
    else
        return false;
}

```

노성민: 나는 생성자 서두에 초깃값을 초기화하는 방식으로 코드를 써봤어. 그리고 이제 bool타입의 함수 isSquare에서 가로와 세로를 비교해서 같으면 true를 반환하고 다르면 false를 반환하게 하면 되게 적었어.

Q5. 최종코드

```

#include<iostream>
using namespace std;

```

```

class Rectangle{
public:
    int width, height;
    bool isSquare();
    Rectangle();
    Rectangle(int x);
    Rectangle(int x, int y);
};

```

```

Rectangle::Rectangle() : Rectangle(1){}

```

```

Rectangle::Rectangle(int x)
    : width(x), height(x){}

```

```

Rectangle::Rectangle(int x, int y) :
    width(x), height(y) {}

```

```

bool Rectangle::isSquare() {

```

```

        if (width == height)
            return true;
        else
            return false;
    }

int main() {
    Rectangle rect1;
    Rectangle rect2(3, 5);
    Rectangle rect3(3);
    if (rect1.isSquare()) cout << "rect1은 정사각형이다." << endl;
    if (rect2.isSquare()) cout << "rect2는 정사각형이다." << endl;
    if (rect3.isSquare()) cout << "rect3는 정사각형이다." << endl;
}

```

Q6. 3-8 지역 객체와 전역 객체의 생성 및 소멸 순서

코드

```

#include <iostream>
using namespace std;
class Circle {
public:
    int radius;
    Circle();
    Circle(int r);
    ~Circle();
    double getArea();
};
Circle::Circle() {
    radius = 1;
    cout << "반지름 " << radius << " 원 생성" << endl;
}
Circle::Circle(int r) {
    radius = r;
    cout << "반지름 " << radius << " 원 생성" << endl;
}
Circle::~Circle() {
    cout << "반지름 " << radius << " 원 소멸" << endl;
}
double Circle::getArea() {
    return 3.14*radius*radius;
}

Circle globalDonut(1000);
Circle globalPizza(2000);

void f() {
    Circle fDonut(100);
    Circle fPizza(200);
}

```

```
int main() {  
    Circle mainDonut;  
    Circle mainPizza(30);  
    f();  
}
```

다음 프로그램의 실행 결과는 무엇인가?

김정호: 위에 설명대로 전역 객체는 프로그램에서 선언된 순서대로 생성되고 지역 객체는 함수가 호출되는 순간에 순서대로 생성된다고 하고, 소멸은 프로그램이 종료하면 전역 객체가 생성된 역순으로 소멸하고, 함수가 종료하면 지역 객체가 생성된 역순으로 소멸한다고 해.

노성민: 그러면 이 코드에서 전역변수는 globalDonut(1000), globalPizza(2000) 이니까 먼저 생성될 것이고 main 함수로 넘어가서 지역 객체들이 생성될 텐데 순서는 mainDonut, mainPizza(30), f() 함수 순서대로 생성되겠네. 그리고 소멸은 역순대로니까 생성의 반대로 소멸 될거야. 결과를 확인해보자.

결과

반지름 1000 원 생성
반지름 2000 원 생성
반지름 1 원 생성
반지름 30 원 생성
반지름 100 원 생성
반지름 200 원 생성
반지름 200 원 소멸
반지름 100 원 소멸
반지름 30 원 소멸
반지름 1 원 소멸
반지름 2000 원 소멸
반지름 1000 원 소멸