

프로그래밍 언어 (2) 실습

13주차

상속

- ex1)

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

// 2차원 평면에서 한 점을 표현하는 클래스 Point
선언
class Point {
    int x, y; //한 점 (x,y) 좌표값
public:
    void set(int x, int y) { this->x = x; this->y = y; }
    void showPoint() {
        cout << "(" << x << ", " << y << ")" << endl;
    }
};
```

```
class ColorPoint : public Point { // 2차원
평면에서 컬러 점을 표현하는 클래스 ColorPoint.
Point를 상속받음
    string color; // 점의 색 표현
public:
    void setColor(string color) { this->color = color;
}
    void showColorPoint();
};

void ColorPoint::showColorPoint() {
    cout << color << ":";
    showPoint(); // Point의 showPoint() 호출
}

int main() {
    Point p; // 기본 클래스의 객체 생성
    ColorPoint cp; // 파생 클래스의 객체 생성
    cp.set(3,4); // 기본 클래스의 멤버 호출
    cp.setColor("Red"); // 파생 클래스의 멤버 호출
    cp.showColorPoint(); // 파생 클래스의 멤버
호출
}
```

상속

• ex1) 계속

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

class Point {
protected:
    int x, y; //한 점 (x,y) 좌표값
public:
    void set(int x, int y);
    void showPoint();
};
void Point::set(int x, int y) {
    this->x = x;
    this->y = y;
}
void Point::showPoint() {
    cout << "(" << x << ", " << y << ")" << endl;
}

class ColorPoint : public Point {
    string color;
public:
    void setColor(string color);
    void showColorPoint();
    bool equals(ColorPoint p);
};

void ColorPoint::setColor(string color) {
    this->color = color;
}
```

```
void ColorPoint::showColorPoint() {
    cout << color << " ";
    showPoint(); // Point 클래스의 showPoint() 호출
}

bool ColorPoint::equals(ColorPoint p) {
    if(x == p.x && y == p.y && color == p.color) // ①
        return true;
    else
        return false;
}

int main() {
    Point p; // 기본 클래스의 객체 생성
    p.set(2,3);
    p.x = 5; // ③ 오류
    p.y = 5; // ④ 오류
    p.showPoint();

    ColorPoint cp; // 파생 클래스의 객체 생성
    cp.x = 10; // ⑤ 오류
    cp.y = 10; // ⑥ 오류
    cp.set(3,4);
    cp.setColor("Red");
    cp.showColorPoint();

    ColorPoint cp2;
    cp2.set(3,4);
    cp2.setColor("Red");
    cout << ((cp.equals(cp2))?"true":"false"); // ⑦
}
```

다중 상속

- ex2)

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

class Adder {
protected:
    int add(int a, int b) { return a+b; }
};

class Subtractor {
protected:
    int minus(int a, int b) { return a-b; }
};
```

```
// 다중 상속
class Calculator : public Adder, public
Subtractor {
public:
    int calc(char op, int a, int b);
};

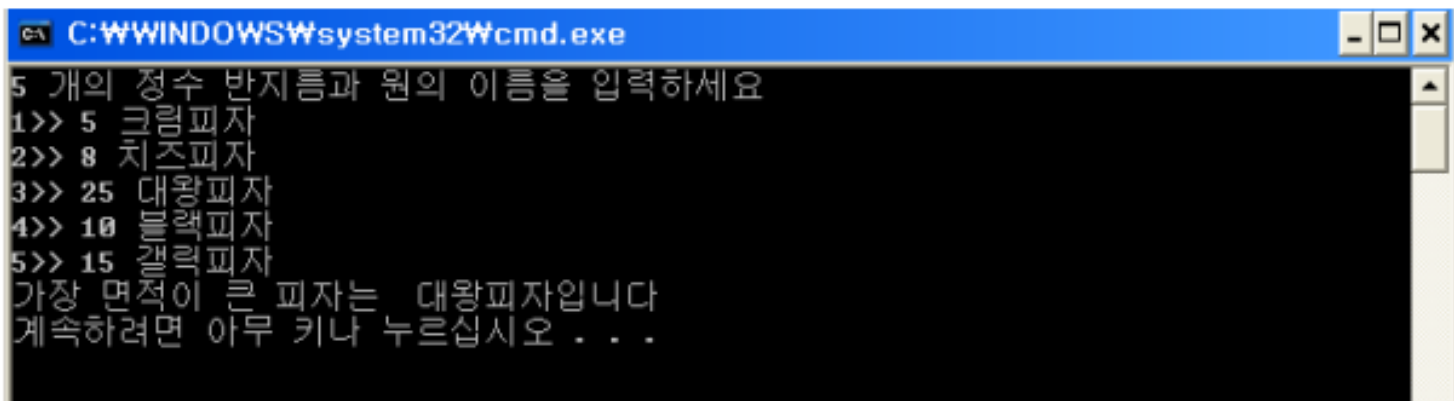
int Calculator::calc(char op, int a, int b) {
    int res=0;
    switch(op) {
        case '+': res = add(a, b); break;
        case '-': res = minus(a, b); break;
    }
    return res;
}
```

```
int main() {
    Calculator handCalculator;
    cout << "2 + 4 = "
        << handCalculator.calc('+', 2, 4) <<
endl;
    cout << "100 - 8 = "
        << handCalculator.calc('-', 100, 8) <<
endl;
}
```

문제

- ex3)

- 다음과 같이 배열을 선언하여 다음 실행 결과가 나오도록 Circle을 상속받는 NamedCircle 클래스와 main 함수 등 필요한 함수를 작성하라.



```
C:\WINDOWS\system32\cmd.exe
5 개의 정수 반지름과 원의 이름을 입력하세요
1>> 5 크림피자
2>> 8 치즈피자
3>> 25 대왕피자
4>> 10 블랙피자
5>> 15 갤럭시피자
가장 면적이 큰 피자는 대왕피자입니다
계속하려면 아무 키나 누르십시오 . . .
```

문제

- ex3) 계속

```
int main() {
    NamedCircle c[5];
    cout << "5 개의 정수 반지름과 원의 이름을 입력하세요" << endl;
    for(int i=0; i<5; i++) {
        int r;
        string name;
        cout << i+1 << ">> ";
        cin >> r;
        getline(cin, name);
        if(r <= 0) {
            cout << "다시 입력하세요" << endl;
            --;
            continue;
        }

        c[i].set(r, name);
    }
    cout << "가장 면적이 큰 피자는 " << biggest(c, 5) << "입니다" << endl;
}
```

Q & A