

11. 가상함수와 추상클래스

[실습 11-1]

- Shape 클래스를 상속받아 타원을 표현하는 Oval, 사각형을 표현하는 Rect, 삼각형을 표현하는 Triangular 클래스를 작성하라. main()을 작성하고 실행하면 다음과 같다.

```
int main() {  
    Shape *p[3];  
  
    p[0] = new Oval("빈대떡", 10, 20);  
    p[1] = new Rect("찰떡", 30, 40);  
    p[2] = new Triangular("토스트", 30, 40);  
  
    for(int i=0; i<3; i++)  
        cout << p[i]→getName() << " 넓이는 " << p[i]→getArea() << endl;  
  
    for(int i=0; i<3; i++) delete p[i];  
}
```

```
빈대떡 넓이는 628  
찰떡 넓이는 1200  
토스트 넓이는 600
```

[실습 11-2]

- 다음 추상 클래스 Calculator를 상속받아 GoodCalc 클래스를 구현하라.

```
class Calculator {  
public:  
    virtual int add(int a, int b) = 0; // 두 정수의 합 리턴  
    virtual int subtract(int a, int b) = 0; // 두 정수의 차 리턴  
    virtual double average(int a [], int size) = 0; // 배열 a의 평균 리턴. size는 배열의 크기  
};
```

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

// 이 곳에 Calculator 클래스 코드 필요

```
class GoodCalc : public Calculator {
```



```
};
```

순수 가상 함수 구현

```
int main() {  
    int a[] = {1,2,3,4,5};  
    Calculator *p = new GoodCalc();  
    cout << p->add(2, 3) << endl;  
    cout << p->subtract(2, 3) << endl;  
    cout << p->average(a, 5) << endl;  
    delete p;  
}
```

```
5  
-1  
3
```

[실습 11-3]

- 다음 코드와 실행 결과를 참고하여 추상 클래스 Calculator를 상속받는 Adder와 Subtractor 클래스를 구현하라

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

class Calculator {
    void input() {
        cout << "정수 2 개를 입력하세요>> ";
        cin >> a >> b;
    }
protected:
    int a, b;
    virtual int calc(int a, int b) = 0; // 두 정수의 합 리턴
public:
    void run() {
        input();
        cout << "계산된 값은 " << calc(a, b) << endl;
    }
};

int main() {
    Adder adder;
    Subtractor subtractor;
    adder.run();
    subtractor.run();
}
```

adder.run()에 의한 실행 결과

정수 2 개를 입력하세요>> 5 3
계산된 값은 8
정수 2 개를 입력하세요>> 5 3
계산된 값은 2

subtractor.run()에 의한 실행
결과