

# 12. 상속, 가상함수와 추상클래스

# 실습내용

---

- 실습#12 : 상속, 가상함수와 추상클래스
- 파일명 : 실습과제#12\_2022199002\_홍길동

## [실습 12-1]

- 다음 코드가 실행되도록 부모클래스 Circle과 자식클래스 NamedCircle 클래스를 작성하고 프로그램을 완성하시오.

```
int main() {  
    NamedCircle waffle(3, "waffle"); // 반지름이 3이고 이름이 waffle인 원  
    waffle.show();  
}
```

반지름이 3인 waffle

## [실습 12-2]

- 다음과 같은 배열을 선언하여 다음 실행결과가 나오도록 부모클래스 Circle과 자식클래스 NamedCircle 그리고 main() 함수 등 필요한 함수를 작성하시오 (함수의 개수는 제한 없음).

```
NamedCircle pizza[5];
```

5개의 정수 반지름과 원의 이름을 입력하세요.

1 >> 5 크림피자

2 >> 8 치즈피자

3 >> 25 대왕피자

4 >> 30 블랙홀피자

5 >> 15 마늘피자

가장 면적이 큰 피자는 블랙홀피자입니다.

## [실습 12-3]

■ 다음과 같은 BaseArray 클래스를 상속받아 큐처럼 작동하는 MyQueue 클래스를 작성하시오.

```
class BaseArray {  
private:  
    int capacity; // 동적 할당된 메모리 용량  
    int *mem; // 정수 배열을 만들기 위한 메모리 포인터  
protected:  
    BaseArray(int capacity=100) {  
        this->capacity = capacity; mem = new int [capacity];  
    }  
    ~BaseArray() { delete [] mem; }  
    void put(int index, int val) { mem[index] = val; }  
    int get(int index) { return mem[index]; }  
    int getCapacity() { return capacity; }  
};
```

```
int main() {  
    MyQueue mQ(100);  
    int n;  
  
    cout << "큐에 삽입할 5개의 정수를 입력하라>> ";  
    for(int i=0; i<5; i++) {  
        cin >> n;  
        mQ.enqueue(n); // 큐에 삽입  
    }  
  
    cout << "큐의 용량 : " << mQ.capacity() << ", 큐의 크기 : " << mQ.length() << endl;  
  
    cout << "큐의 원소를 순서대로 제거하여 출력한다>> ";  
    while(mQ.length() != 0) {  
        cout << mQ.dequeue() << ' '; // 큐에서 제거하여 출력  
    }  
  
    cout << endl << "큐의 현재 크기 : " << mQ.length() << endl;  
}
```

큐에 삽입할 5개의 정수를 입력하라>> 1 3 5 7 9

큐의 용량 : 100, 큐의 크기 : 5

큐의 원소를 순서대로 제거하여 출력한다>>1 3 5 7 9

큐의 현재 크기 : 0.

## [실습 12-4]

- Shape 클래스를 상속받아 타원을 표현하는 Oval, 사각형을 표현하는 Rect, 삼각형을 표현하는 Triangular 클래스를 작성하시오. main()을 작성하고 실행하면 다음과 같다.
- Shape 클래스는 추상클래스로 만들 것, 가상함수를 사용할것

```
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 12-5]

- 다음 추상 클래스 Calculator를 상속받아 GoodCalc 클래스를 구현하시오.
- GoodCalc 클래스 안에 add, subtract, average 구현 할 것

```
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 {  
};
```

순수가상함  
수 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 12-6]

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

```
#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 의  
한 실행 결과