

cpp_09_상속

☐ 개념 확인 학습

1. 다음 질문에 O, X로 답하세요.

- A. 기본 클래스로 생성된 객체는 파생 클래스 멤버에 접근할 수 있다. ()
- B. 접근 지정자 `protected`로 선언된 멤버는 전역 함수와 외부클래스에서 접근할 수 있다. ()
- C. 상속 접근 지정자가 생략되면 `public`이다. ()
- D. C++는 다중 상속을 허용한다. ()
- E. 파생클래스 생성자에서 명시적으로 기본 클래스 생성자를 선택하지 않으면 묵시적으로 기본 생성자를 호출한다. ()
- F. 상속은 클래스를 선언할 때 `extends` 키워드를 사용한다. ()
- G. 파생 클래스의 포인터가 기본클래스의 포인터에 치환되는 것을 다운캐스팅 이라한다. ()
- H. 업 캐스팅은 반드시 강제 타입 변환이 필요하다. ()
- I. 파생클래스에서는 상속을 사용하여 기본클래스의 모든 멤버를 사용할 수 있다. ()
- J. `private` 상속은 기본클래스의 `protected`, `public` 멤버를 `private`으로 계승한다. ()

2. C++ 상속의 종류에 대해 설명하고 예를 제시하세요.

3. 다중 상속의 문제점과 해결방법의 예를 제시하고 설명하세요.

4. 가상 상속을 선언할 때 사용하는 키워드는 무엇인가요?

5. 클래스 `Rocket`과 클래스 `Computer`를 동시에 상속받는 클래스 `Statellite`를 선언하는 문장을 작성하세요.

6. 다음 코드를 사용해 객체 b가 생성될 때 화면에 출력되는 내용은 무엇인가요?

```
class A {
public:
    A() { cout << "생성자 A" << endl; }
    A(int x) { cout << "생성자 A" << x << endl; }
};
class B: public A {
public:
    B() { cout << "생성자 B" << endl; }
    B(int x) { cout << "생성자 B" << x << endl; }
    B(int x, int y) : A(x+y+2) { cout << "생성자 B" << x + y + 2 << endl; }
};
```

- A. B b;
- B. B b(10);
- C. B b(10, 20);

☐ 응용 프로그래밍

7. BaseArray클래스를 상속 받아 큐로 작동하는 MyQueue 클래스를 작성하세요.

```
class BaseArray {
    int capacity; //동적 할당된 메모리의 총 용량
    int *mem; //동적 할당된 메모리
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; }
};

class MyQueue : public BaseArray {
    int head; //데이터를 넣을 위치는 head
    int tail; //데이터 꺼내는 위치는 tail+1
    int size; //현재 큐 안에 있는 데이터의 개수
public:
    MyQueue(int capacity); //capacity만큼 BaseArray 생성, head, tail, size 초기화
    void enqueue(int n); //큐의 head 위치에 데이터 넣기
    int dequeue(); //큐의 tail+1 위치에서 데이터를 꺼내서 리턴
};
```

```

int capacity(); //큐의 총 용량 리턴
int length(); //큐에 있는 데이터의 개수 리턴
};

int main() {
    int capa;
    cout << "큐의 전체 용량은? ";
    cin >> capa;
    MyQueue mq(capa);

    int cnt, n;
    cout << "큐에 저장할 정수 개수는? ";
    cin >> cnt;
    for (int i = 0; i < cnt; i++) {
        cout << i << " ";
        cin >> n;
        mq.enqueue(n); //큐에 데이터 넣기
    }
    cout << "큐의 용량 = " << mq.capacity() << endl;
    cout << "큐의 현재 크기 = " << mq.length() << endl;

    cout << "큐의 원소를 순서대로 꺼내 출력 하면...";
    while (mq.length() != 0) {
        cout << mq.dequeue() << ' '; //큐에서 꺼내 출력
    }
    cout << endl;
    cout << "큐의 용량 = " << mq.capacity() << endl;
    cout << "큐의 현재 크기 = " << mq.length() << endl;
}

```

```

큐의 전체 용량은? 30
큐에 저장할 정수 개수는? 3
0) 5
1) 6
2) 7
큐의 용량 = 30
큐의 현재 크기 = 3
큐의 원소를 순서대로 꺼내 출력 하면...5 6 7
큐의 용량 = 30
큐의 현재 크기 = 0

```