

07. 여러가지 객체의 생성방법

[실습 7-1]

- 다음과 같이 원을 추상화한 Circle 클래스가 있다. Circle 클래스와 main() 함수를 작성하고 3개의 Circle 객체를 가진 배열을 선언하고, 반지름 값을 입력받고 면적이 100보다 큰 원의 개수를 출력하는 프로그램을 완성하라. Circle 클래스도 완성하라.

```
class Circle {  
    int radius; // 원의 반지름 값  
public:  
    void setRadius(int radius); // 반지름을 설정한다.  
    double getArea(); // 면적을 리턴한다.  
};
```

```
원 1의 반지름 >> 5  
원 2의 반지름 >> 6  
원 3의 반지름 >> 7  
면적이 100보다 큰 원 2개입니다.
```

[실습 7-2]

- 다음은 색의 3요소인 red, green, blue로 색을 추상화한 Color 클래스를 선언하고 활용하는 코드이다. 빈칸을 채워라.
- 여기서 red, green, blue는 0~255의 값만 가진다.

```
class Color {  
    int red, green, blue;  
public:  
    Color() { red = green = blue = 0; }  
    Color(int r, int g, int b) { red = r; green = g; blue = b; }  
    void setColor(int r, int g, int b) { red = r; green = g; blue = b; }  
    void show() { cout << red << ' ' << green << ' ' << blue << endl; }  
};
```

```
int main() {  
    Color screenColor(255, 0, 0); // 빨간색의 screenColor 객체 생성  
    Color *p;                      // Color 타입의 포인터 변수 p 선언  
    _____;                  // (1) p가 screenColor의 주소를 가지도록 코드 작성  
    _____;                  // (2) p와 show()를 이용하여 screenColor 색 출력  
    _____;                  // (3) Color의 일차원 배열 colors 선언. 원소는 3개  
    _____;                  // (4) p가 colors 배열을 가리키도록 코드 작성  
  
    // (5) p와 setColor()를 이용하여 colors[0], colors[1], colors[2]가  
    // 각각 빨강, 초록, 파랑색을 가지도록 코드 작성  
    _____;  
    _____;  
    _____;  
  
    // (6) p와 show()를 이용하여 colors 배열의 모든 객체의 색 출력. for 문 이용  
    _____;  
    _____;  
    _____;  
    _____;  
}
```

```
255 0 0  
255 0 0  
0 255 0  
0 0 255
```

[실습 7-3]

- 다음에서 Person은 사람을, Family는 가족을 추상화한 클래스로서 완성되지 않은 클래스이다.

```
class Person {
    string name;
public:
    Person() { name=""; }
    Person(string name) { this->name = name; }
    string getName() { return name; }
    void setName(string name) { this->name = name; }
};

class Family {
    string name;
    Person* p; // Person 배열 포인터
    int size; // Person 배열의 크기. 가족 구성원 수
public:
    Family(string name, int size); // size 개수만큼 Person 배열 동적 생성
    void setName(int index, string name);
    void show(); // 모든 가족 구성원 출력
    ~Family();
};
```

[실습 7-3] 계속

- 다음 main()이 작동되도록 Person과 Family 클래스에 필요한 멤버들을 추가하고, 코드를 완성하라.

```
int main() {  
    Family *simpson = new Family("Simpson", 3); // 3명으로 구성된 Simpson 가족  
    simpson->setName(0, "Mr. Simpson");  
    simpson->setName(1, "Mrs. Simpson");  
    simpson->setName(2, "Bart Simpson");  
    simpson->show();  
    delete simpson;  
}
```

Simpson 가족은 다음과 같이 3명입니다.
Mr. Simpson Mrs. Simpson Bart Simpson