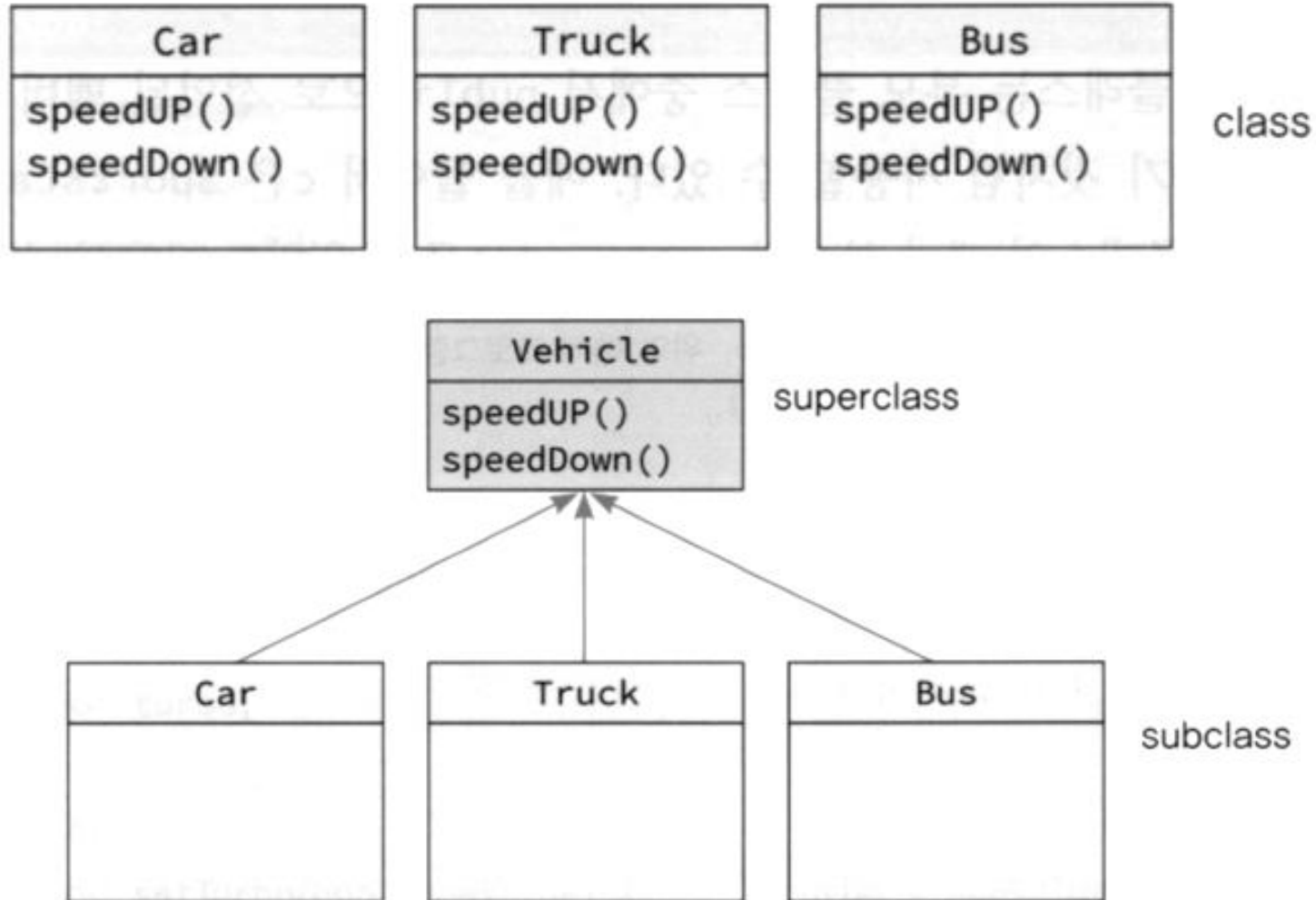


상속

상속

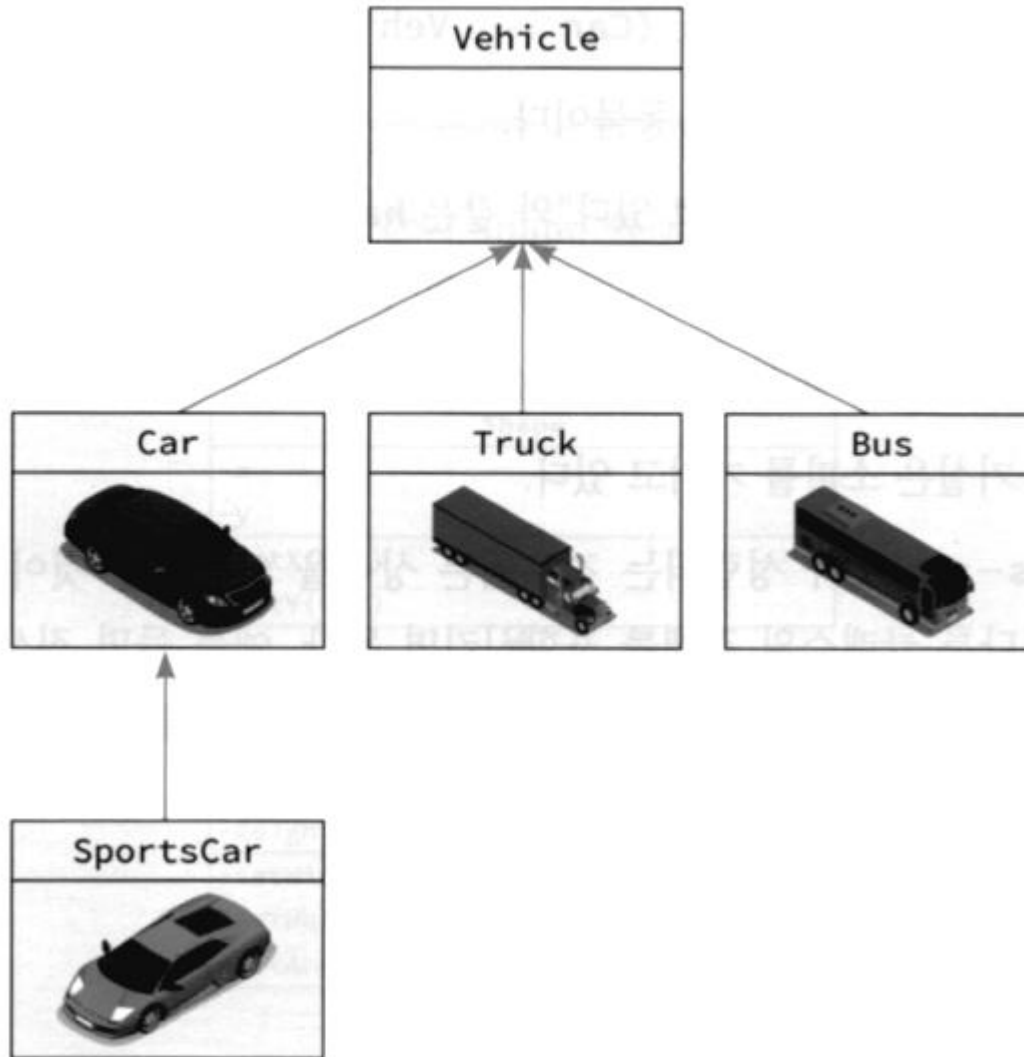
■ 상속의 필요성

- 코드 중복



상속

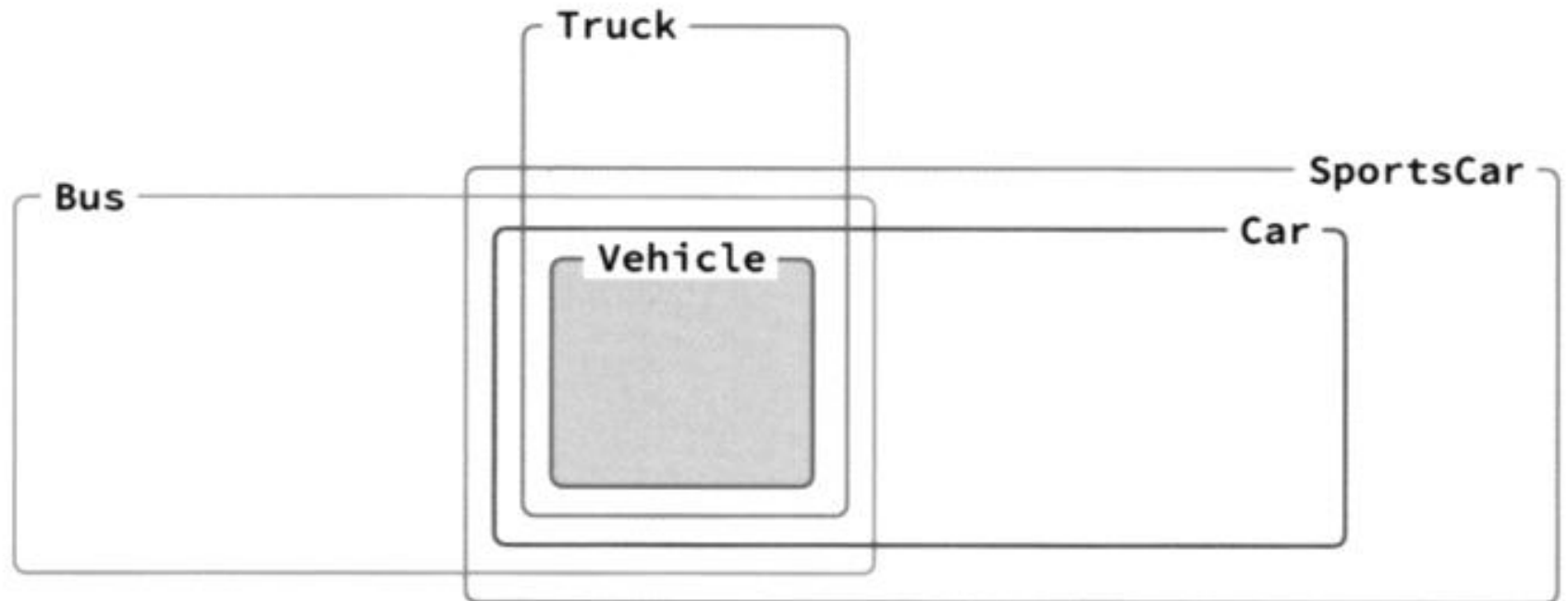
■ 상속 계층 구조도



상속

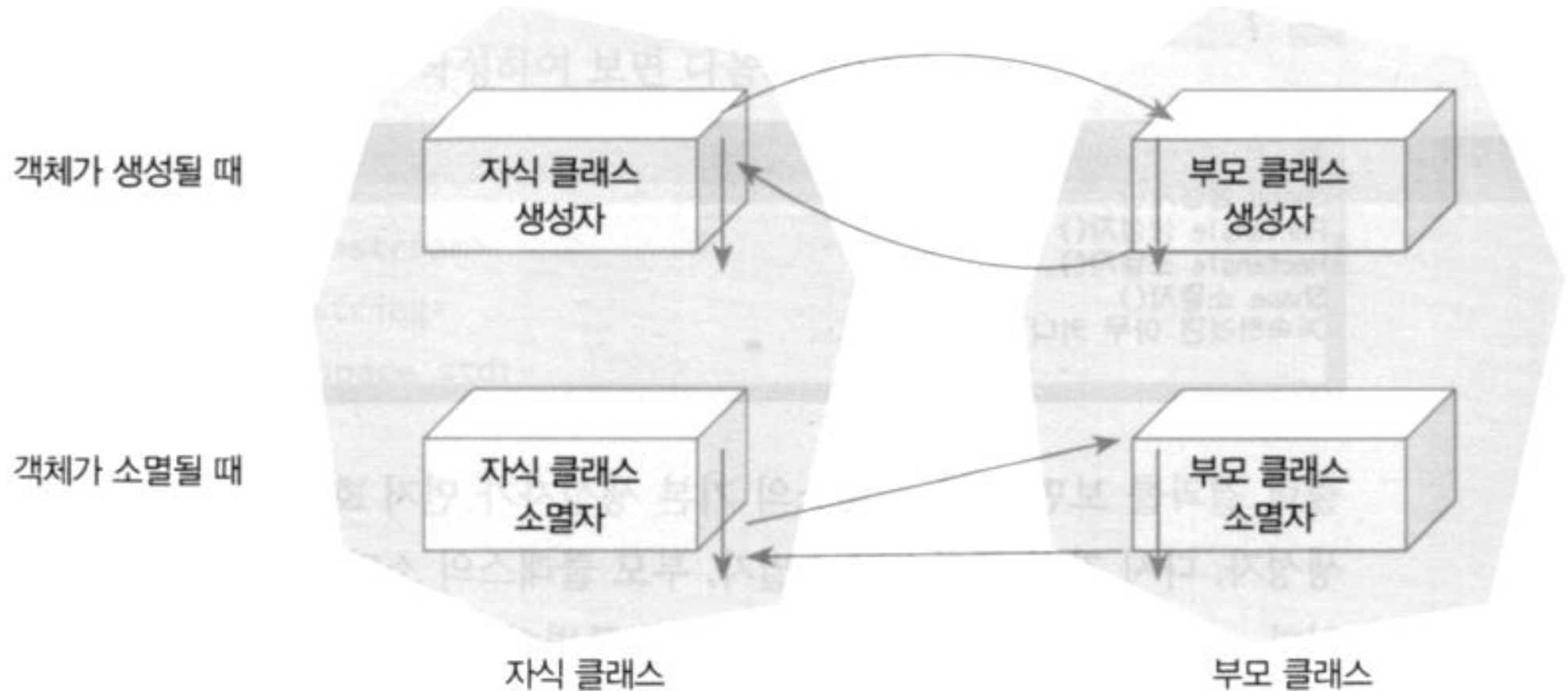
■ 상속 계층 구조도

```
class Vehicle {    ...    }  
class Car : public Vehicle { ... }  
class Truck : public Vehicle { ... }  
class Bus : public Vehicle { ... }  
class SportsCar : public Car { ... }
```



상속

■ 상속에서의 생성자, 소멸자



■ 부모 클래스의 생성자를 지정하는 방법

자식클래스의 생성자() : 부모클래스의 생성자()

{

}

■ chapter09/ex01_inherit.cpp] 부모 클래스의 생성자를 지정하는 방법

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

class Shape {
    int x, y;
public:
    Shape() {
        cout << "Shape() 생성자" << endl;
    }

    Shape(int xloc, int yloc): x(xloc), y(yloc) {
        cout << "Shape(xloc, yloc) 생성자" << endl;
    }

    ~Shape() {
        cout << "~Shape() 소멸자" << endl;
    }
};
```

■ chapter09/ex01_inherit.cpp] 부모 클래스의 생성자를 지정하는 방법

```
class Rectangle: public Shape {
    int width, height;

public:
    Rectangle() {
        cout << "Rectangle() 생성자" << endl;
    }

    Rectangle(int x,int y, int w, int h): Shape(x, y), width(w), height(h) {
        cout << "Rectangle(x, y, w, h) 생성자" << endl;
    }

    ~Rectangle() {
        cout << "~Rectangle() 소멸자" << endl;
    }
};
```


■ chapter09/ex01_inherit.cpp] 부모 클래스의 생성자를 지정하는 방법

```
int main(int argc, char const *argv[])
{
    Rectangle r1;
    cout << endl;
    Rectangle r2(0, 0, 100, 100);
    cout << endl;

    return 0;
}
```

Shape() 생성자

Rectangle() 생성자

Shape(xloc, yloc) 생성자

Rectangle(x, y, w, h) 생성자

~Rectangle() 소멸자

~Shape() 소멸자

~Rectangle() 소멸자

~Shape() 소멸자

멤버 함수 재정의(override)

■ 멤버 함수 재정의

- 부모 클래스의 멤버 함수와 모양(원형)이 동일해야 함
- 다르면 새로운 멤버 함수로 오버로드 됨



멤버 함수 재정의(override)

■ chapter09/ex2_override.cpp] 멤버 함수 재정의

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

class Animal {
public:
    void speak() {
        cout << "동물이 소리를 내고 있음" << endl;
    }
};

class Dog : public Animal {
public:
    void speak() {
        cout << "개가 소리를 내고 있음" << endl;
    }
};
```

멤버 함수 재정의(override)

■ chapter09/ex2_override.cpp] 멤버 함수 재정의

```
int main(int argc, char const *argv[])
{
    Dog dog;
    dog.speak();

    return 0;
}
```

개가 소리를 내고 있음

멤버 함수 재정의(override)

■ 부모 클래스의 멤버 함수 호출

- 멤버 함수 재정의 시 부모 클래스의 멤버 함수 호출이 필요한 경우
- 부모 클래스::함수명() 으로 호출

멤버 함수 재정의(override)

■ chapter09/ex3_override.cpp] 부모 클래스의 멤버 함수 호출

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

class SuperClass {
public:
    void print() {
        cout << "SuperClass의 print()" << endl;
    }
};

class ChildClass : public SuperClass {
public:
    void print() {
        SuperClass::print();
        cout << "ChildClass의 print()" << endl;
    }
};
```

멤버 함수 재정의(override)

■ chapter09/ex3_override.cpp] 부모 클래스의 멤버 함수 호출

```
int main(int argc, char const *argv[])
{
    ChildClass c;
    c.print();
    return 0;
}
```

SuperClass의 print()

ChildClass의 print()

다중 상속

■ 다중 상속

```
class Sub : public Sup1, public Sup2
{
    ...// 추가된 멤버
    ...// 재정의된 멤버
}
```

