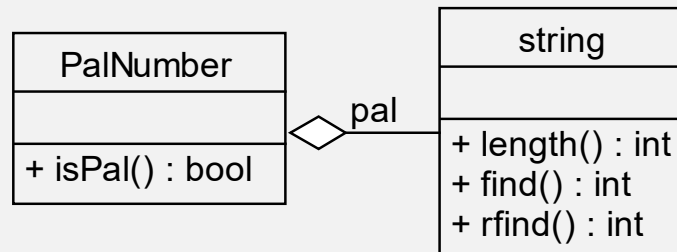




USING AGGREGATION: WHOLE-PART WITH POINTERS

The whole sends messages to (i.e., calls functions in) its parts

USING SIMPLE AGGREGATION

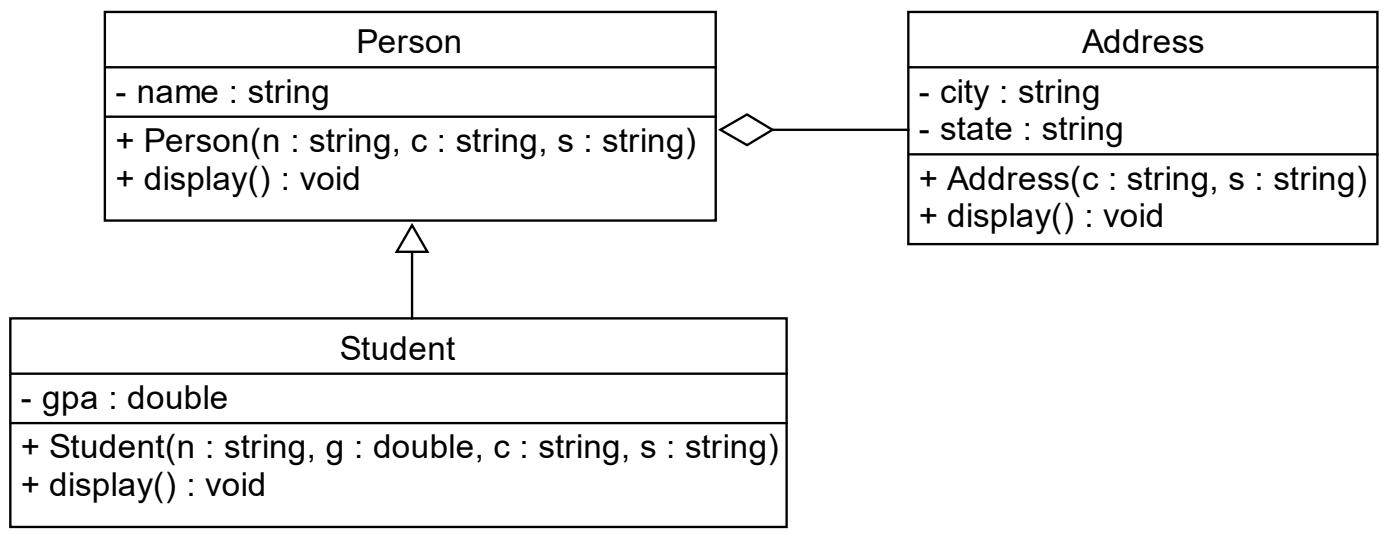


```
class string
{
    public:
        int length() { ... }
        int find() { ... }
        int rfind() { ... }
};
```

```
class PalNumber
{
    private:
        string* pal;
    public:
        bool isPal()
        {
            pal->length() ...
            pal->find() ...
            pal->rfind() ...
        }
};
```



INHERITANCE & AGGREGATION(I)





```
class Address
{
    public:
        void display()
        {
            cout << city << ", " << endl;
        }
};

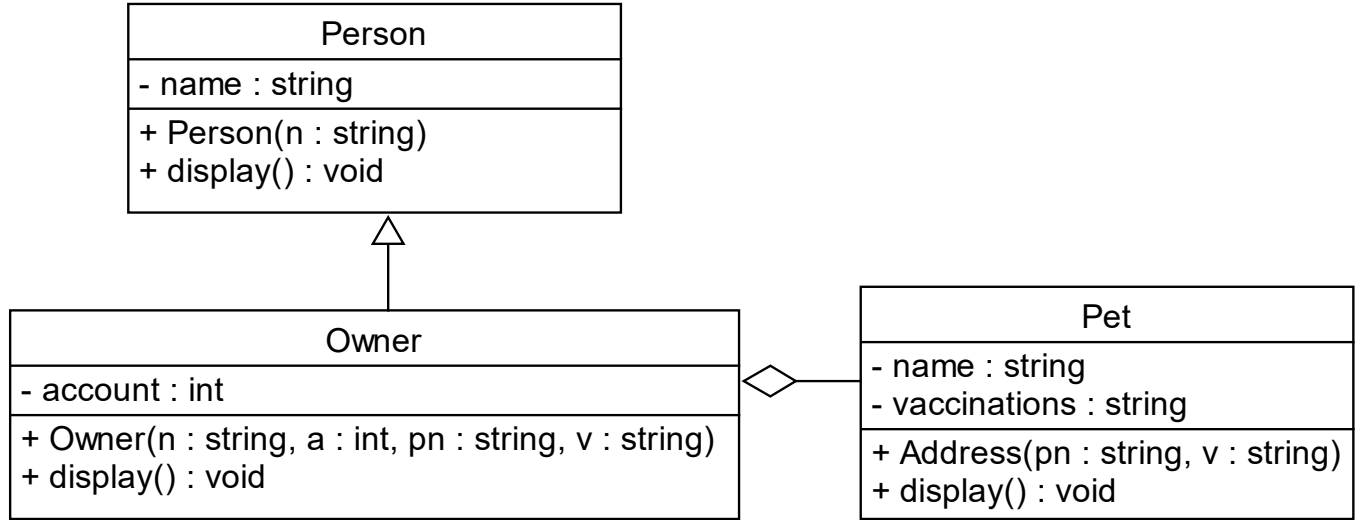
class Person
{
    private:
        Address* addr;
    public:
        void display()
        {
            cout << name << endl;
            if (addr != nullptr)
                addr->display();
        }
};
```

USING AGGREGATION WITH INHERITANCE (I)

```
class Student : public Person
{
    public:
        void display()
        {
            Person::display();
            cout << gpa << endl;
        }
};
```



INHERITANCE & AGGREGATION(2)





USING AGGREGATION WITH INHERITANCE (2)

```
class Pet
{
public:
    void display()
    {
        cout << name << " vaccinated on "
            << vaccinations << endl;
    }
};

class Person
{
public:
    void display()
    {
        cout << name << endl;
        cout << phone << endl;
    }
};
```

```
class Owner : public Person
{
private:
    Pet* my_pet;
public:
    void display()
    {
        Person::display();
        cout << account << endl;
        if (my_pet != nullptr)
            my_pet->display();
    }
};
```