

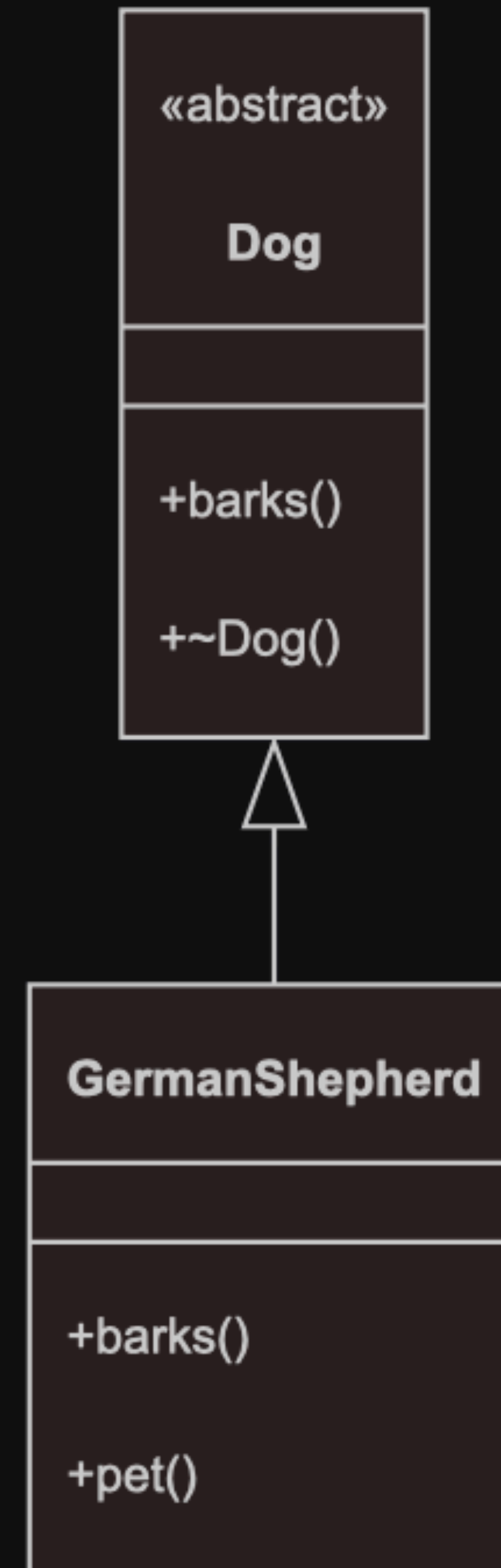
Demo Interface fun with Dogs

```
#include <iostream>
```

```
class Dog {  
public:  
    virtual void barks() = 0;  
    virtual ~Dog() {}  
};
```

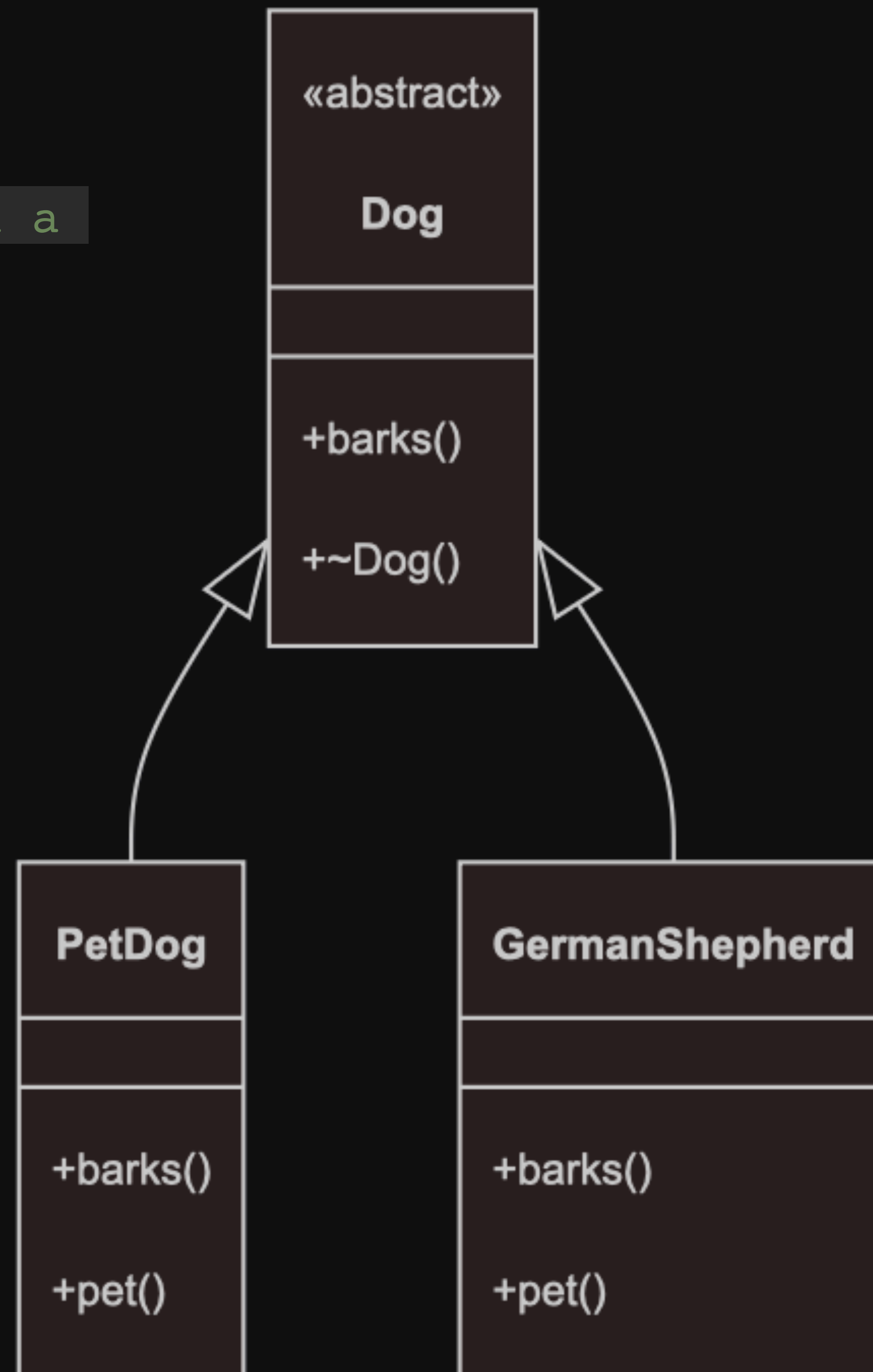
```
class GermanShepherd : public Dog {  
public:  
    void barks() override {  
        std::cout << "WOOF! WOOF! (loud  
and intimidating bark)\n";  
    }  
};
```

```
    void pet() {  
        std::cout << "Rex doesn't like  
to be petted by strangers.\n";  
    }  
};
```



```
class PetDog : public Dog {  
public:  
    void barks() override {  
        std::cout << "Woof! Woof! (with a  
happy wagging tail)\n";  
    }  
};
```

```
    void pet() {  
        std::cout << "*Wags tail  
enthusiastically*\n";  
    }  
};
```



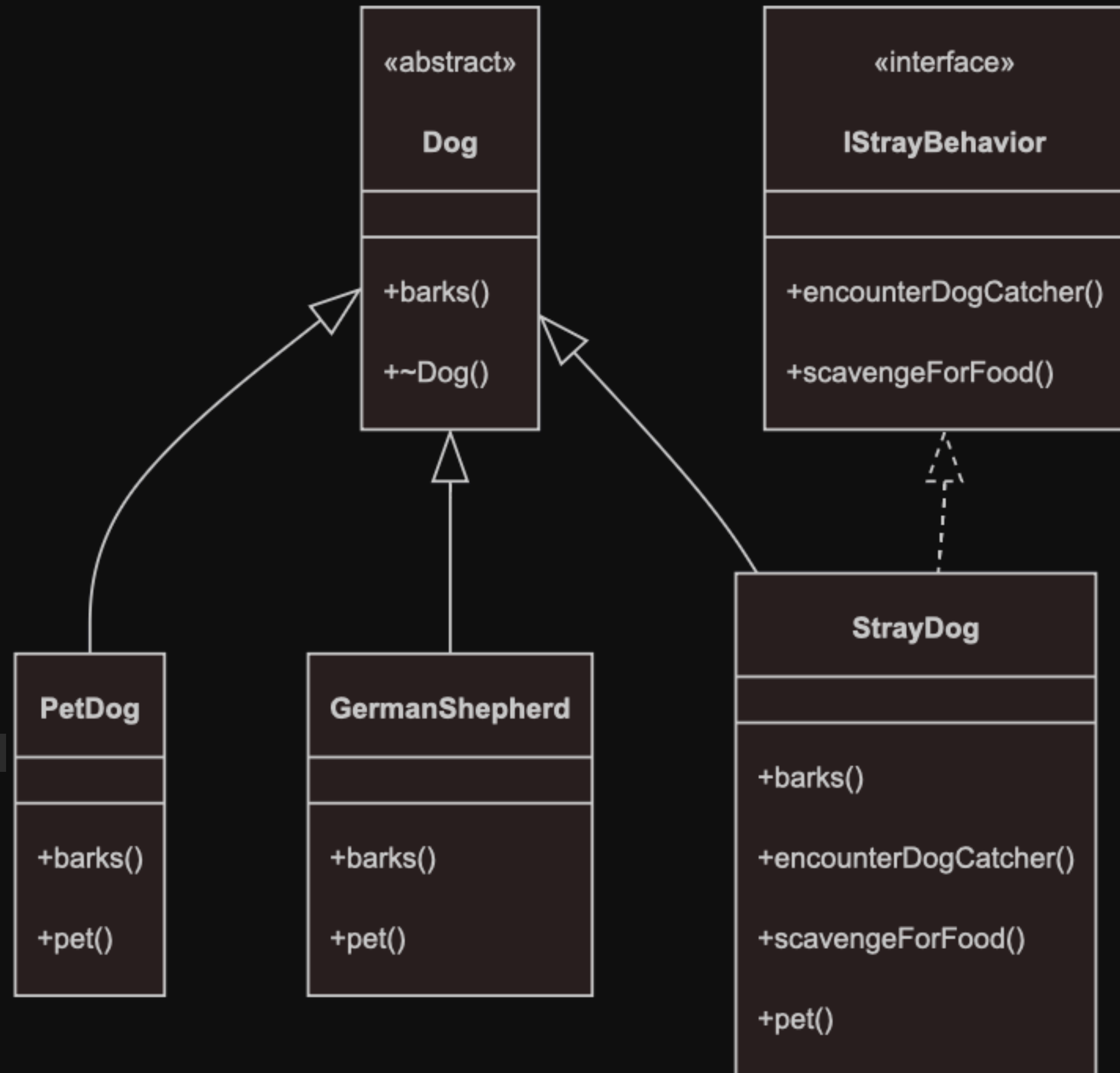
```
class IStrayBehavior {
public:
    virtual void encounterDogCatcher() = 0;
    virtual void scavengeForFood() = 0;
};
```

```
class StrayDog : public Dog, public
IStrayBehavior {
public:
    void barks() override {
        std::cout << "*Nervous yelp*\n";
    }
```

```
    void encounterDogCatcher() override {
        std::cout << "*Panics and flees in
terror*\n";
    }
```

```
    void scavengeForFood() override {
        std::cout << "*Searches for scraps in
the garbage*\n";
    }
```

```
    void pet() {
        std::cout << "*Cowers in fear*\n";
    }
};
```



One more Dog...

- Do all allow us to pet them?!
- Why do all Dogs implement

```

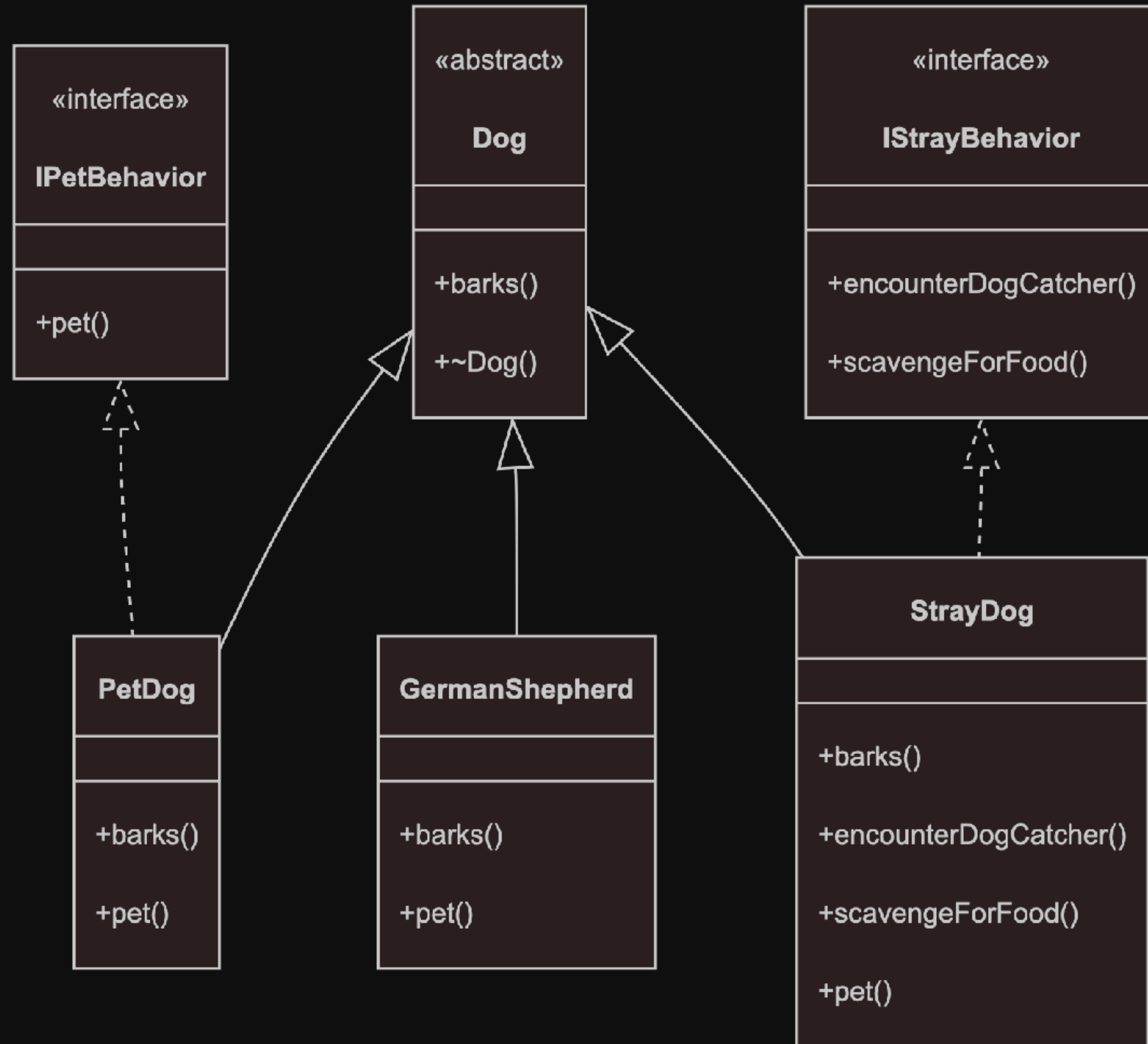
#include <iostream>

// Core dog behavior
class Dog {
public:
    virtual void barks() = 0;
    virtual ~Dog() {}
};

// Stray-specific behavior
class IStrayBehavior {
public:
    virtual void encounterDogCatcher() = 0;
    virtual void scavengeForFood() = 0;
};

// Pet-specific behavior (if needed)
class IPetBehavior {
public:
    virtual void pet() = 0;
};

```



One more Dog...

- Would a stray pet dog German Shepard be possible?!