

Семинар №11

ФАКТ 2021

Бирюков В. А.

November 19, 2021

Полиморфизм

Полиморфизм: Структура базового класса

```
class Animal {
protected:
    std::string mName;
    int mAge;
public:
    Animal(std::string name, int age)
        : mName(name), mAge(age) {}

    void say() const {
        cout << "<abstract sounds>" << endl;
    }
    void move() const {
        cout << "<abstract walking>" << endl;
    }
};
```

Полиморфизм: Структура класса наследника

```
class Cat : public Animal {
private:
    int mMiceCaught;
public:
    Cat(std::string name, int age, int miceCaught)
        : Animal(name, age), mMiceCaught(miceCaught)
    {}

    void say() const {
        cout << "Meow!" << endl;
    }
    void move() const {
        cout << "<sneaking>" << endl;
    }
};
```

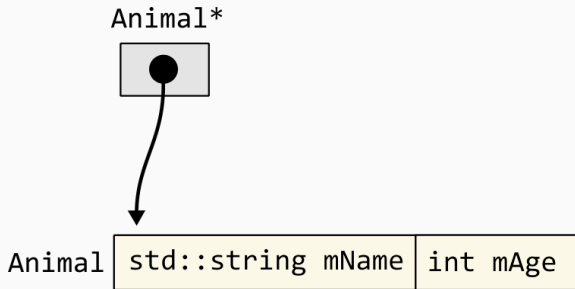
Animal

| | |
|--------------------------------|-----------------------|
| <code>std::string mName</code> | <code>int mAge</code> |
|--------------------------------|-----------------------|

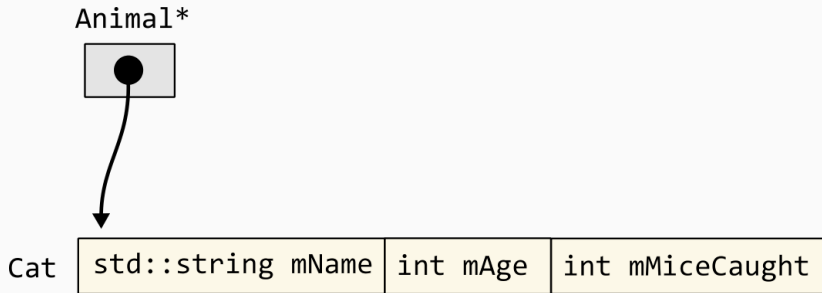
Cat

| | | |
|--------------------------------|-----------------------|------------------------------|
| <code>std::string mName</code> | <code>int mAge</code> | <code>int mMiceCaught</code> |
|--------------------------------|-----------------------|------------------------------|

Указатель базового класса на объект базового класса



Указатель базового класса на объект класса наследника

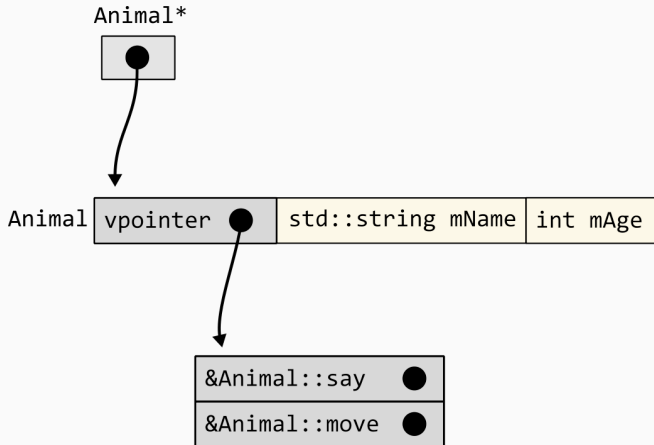


Добавим виртуальные
функции

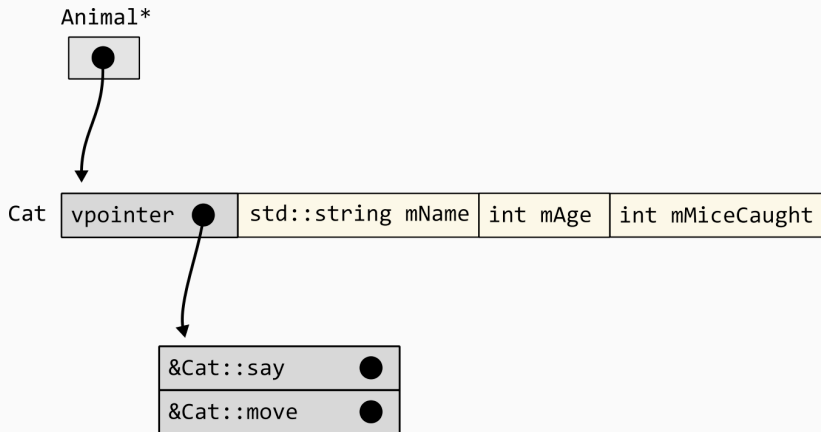
- Реализация полтморфизма не задаётся стандартом
- Но почти всегда используется таблица виртуальных функций

```
class Animal {  
protected:  
    std::string mName;  
    int mAge;  
public:  
    Animal(std::string name, int age)  
        : mName(name), mAge(age) {}  
  
    virtual void say() const {  
        cout << "<abstract sounds>" << endl;  
    }  
    virtual void move() const {  
        cout << "<abstract walking>" << endl;  
    }  
};
```

Указатель базового класса на объект базового класса



Указатель базового класса на объект класса наследника



```
Cat cat {"Cleo", 10, 100};  
Animal* p = &cat;
```

- Статический тип `p` – указатель на `Animal`
- Динамический тип `p` – указатель на `Cat`