

Наследование в ООП

```
class Animal {  
public
```

```
    string name;
```

```
public
```

```
    animal (string new_name = ""): name  
    (new_name) {};
```

```
    sound {};
```

```
}
```

наследование от Animal

```
class Dog : public Animal {  
public:
```

```
    string type;
```

```
    Dog (string new_name = " ",
```

```
        sound {};
```

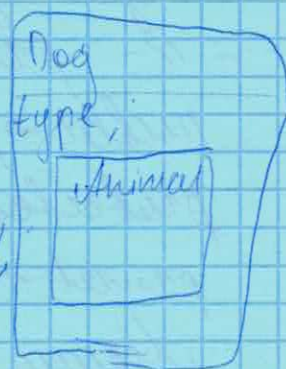
```
};
```

наследует
по умолчанию

Animal



Dog



0 При соз. объектов явно выз. констр.
родителя (в нашем случае Animal)

Если есть наследование, то обяз. соз.

номер. def. атрибуты

Но можно безвозврат и непереносим
комментарий:

Post string user name "11". Animal { }

Однако сейчас

Public - g. шарженое животное

Protected - - " -

Private - только внутри класса

Наследование:

Class Dog: Public Animal {

Множеством методов private Animal

- " - : private Animal

public -> private

protected -> private

private -> наследник

Можно в наследия переименовать уровень

доступа

Class Dog: public Animal {

- " -

Protected

using Animal : : sound

Мелко

sound(1) = bobk

"уражение моего
ушища пог"

пог bobk / "bobk" /

Animal creature / "xros" /

- " -

creature = bobk, вперемешку creature
мониторит также перемешку bobk,
некоторые свойства animal

Animal & creature 01 = 2 bobk, - угад

Animal & creature 02 = bobk, - угад

bobk. sound(1), - из go-pog

creature 01 → sound(1), из animal

creature 02 → sound(1), из sound(1), из animal

Каждо предпринимает угад из noise

Animal как угадывает на noise пог

как из шумовых угад на noise пог

пог bobk / "bobk" /

Animal & creature = 2 bobk,

/ dynamic - last < poq > (creature) / sound(1) /

большой шумов

большой sound(1)
из noise пог

Вирт. функции

Вирт. функции будут общими для всех, и для
каждого будет виртуальное у всех по-
-разному

Вирт. функции - пойдут вирт. (индивидуальная
работа)

Виртуализатор - виртуальная машина есть
распределение

virtual source / source final

присоединяется для
моделирования

Source... final: public... }

закрыт класс - это функция Рес

Обратный класс

Идея: созд. методов и переменных по бу-
дущим результатам

class Virtual {

- II -

virtual void source() + 0 1 функция для
обратного

Классовый класс-класс у пользователя есть
класс бы один активный. фр-ция.

Множественное наследие

Animal
Lion
Tiger → ^{virtual} class Tiger: public Lion, public Tiger {

Проблемы:

Дублирование кода и вызовов метода

Решение: Нам не генерировать повторно
или создать Animal - virtual
class Animal {

- 11 -

{
class Tiger: virtual public Animal {

{
class Tiger: public Lion, public Tiger {



⑩ Dijkstra's algorithm of graph traversal

Class G of
private

Node * root;
Public

Node * Search / Node * not 1.

Class Node of
private

void * data

8th List of Node * > neighbors;

friend class Graph / template by number
to input degree

Node is friend of Node

Ho by Node private manage a node to
neighborhood

A → B

B → C

A no friend C

Class 6 {

- 11 -

Note * search (Note * note)

class Note {

- 1 -

friend Note * G : search (Note * note)

{
 группа, м.л. группы, private
 new Note

(1) Аномальная ситуация

Рег ("Bobik"),

Вспомогательная

Рег ("Bobik") | search()