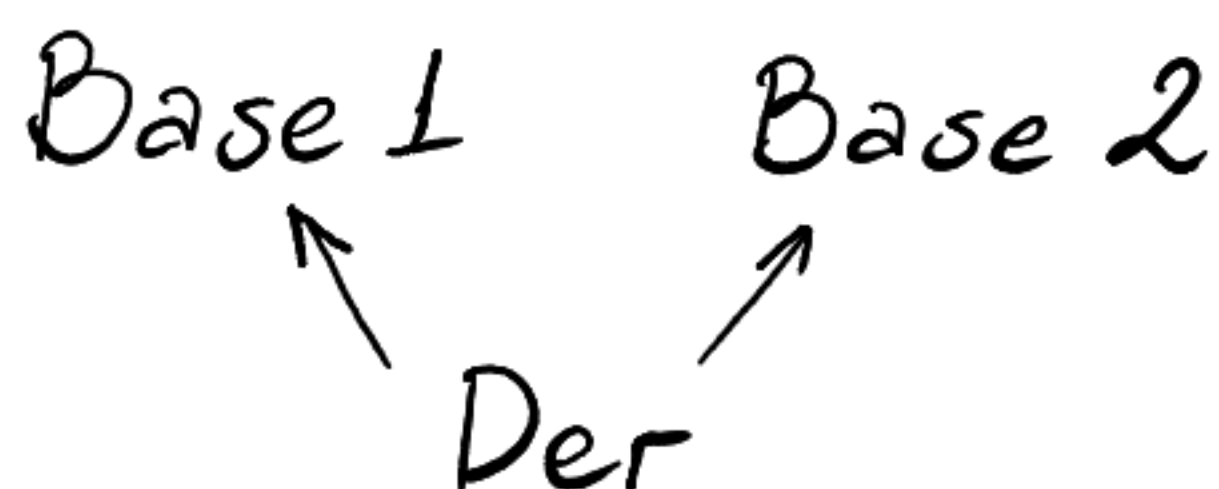


## Тема 12

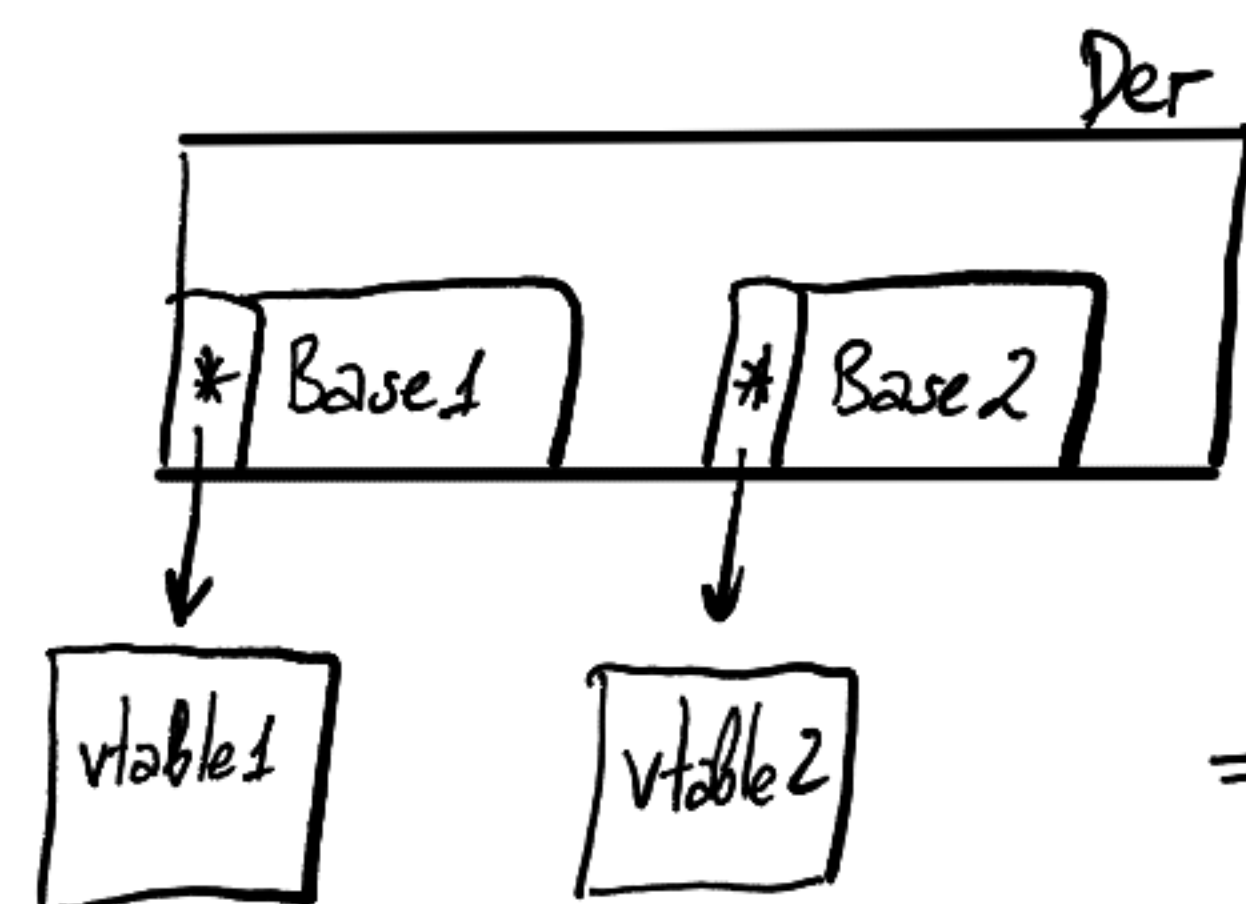
Множествено наследяване. Диамантен проблем.  
Колекции от обекти в полиморфна йерархия.  
Копиране и триене.  
Пример с ферма.

### Множествено наследяване:



$\text{Base 1}^* \text{ ptr1} = \&d;$   
 $\text{Base 2}^* \text{ ptr2} = \&d;$

$\text{ptr1} \neq \text{ptr2}$



$\Rightarrow$  2 базови класа,  
2 виртуални таблици

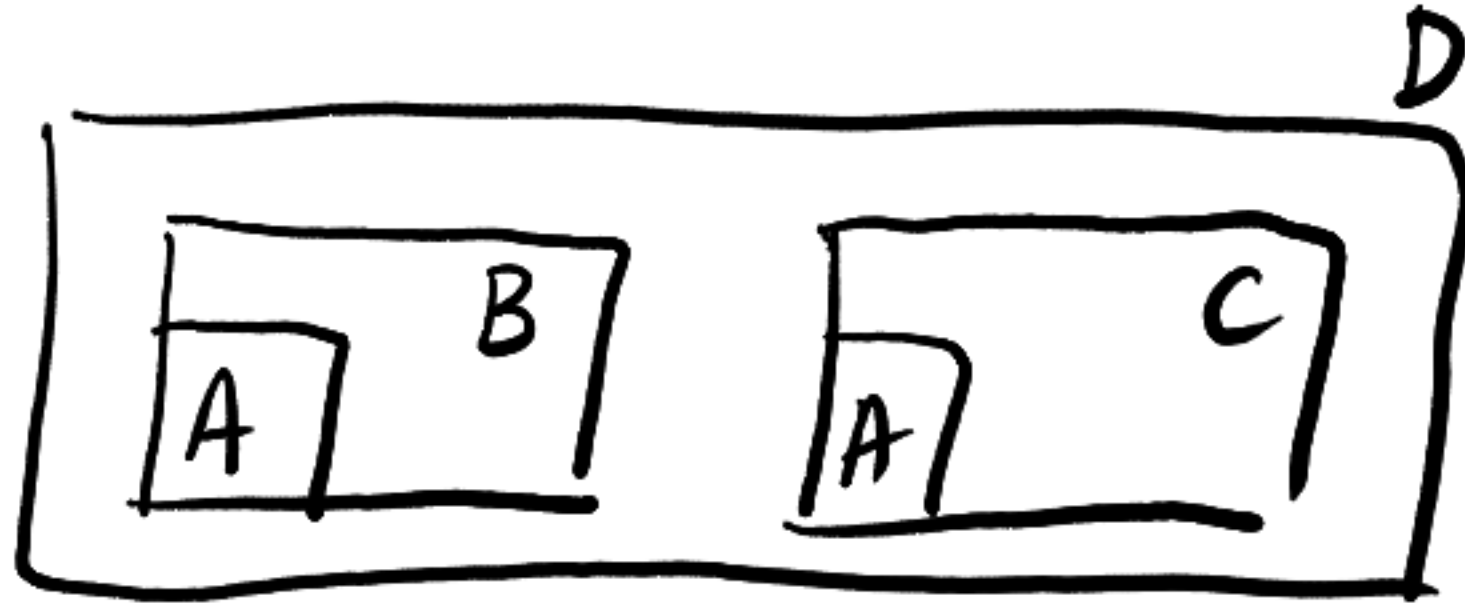
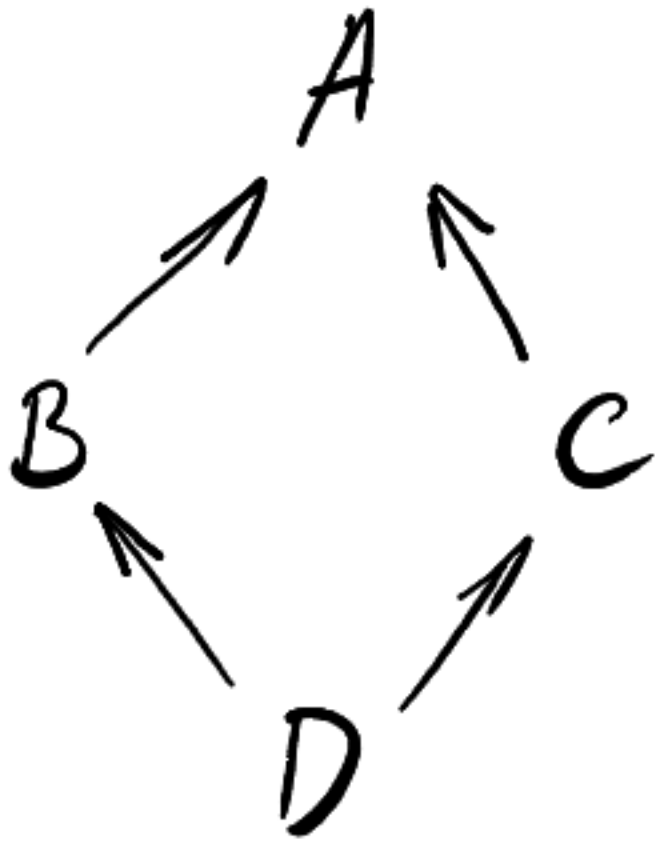
— Таблиците имат параметър  $\Delta$  — какво отместване трябва да направи указателят, за да намери адреса на Der (отстъп от началото на Der до началото на Base)

$$\Delta(\text{Base 1}) = 0$$

$$\Delta(\text{Base 2}) = \text{sizeof}(\text{Base 2})$$

- конструкторът и деструкторът викат  
конструктори и деструктори на Base класовете

## Диамантен проблем



Проблемът е, че може да се получи двусмислица,  
заради двете инстанции от клас A

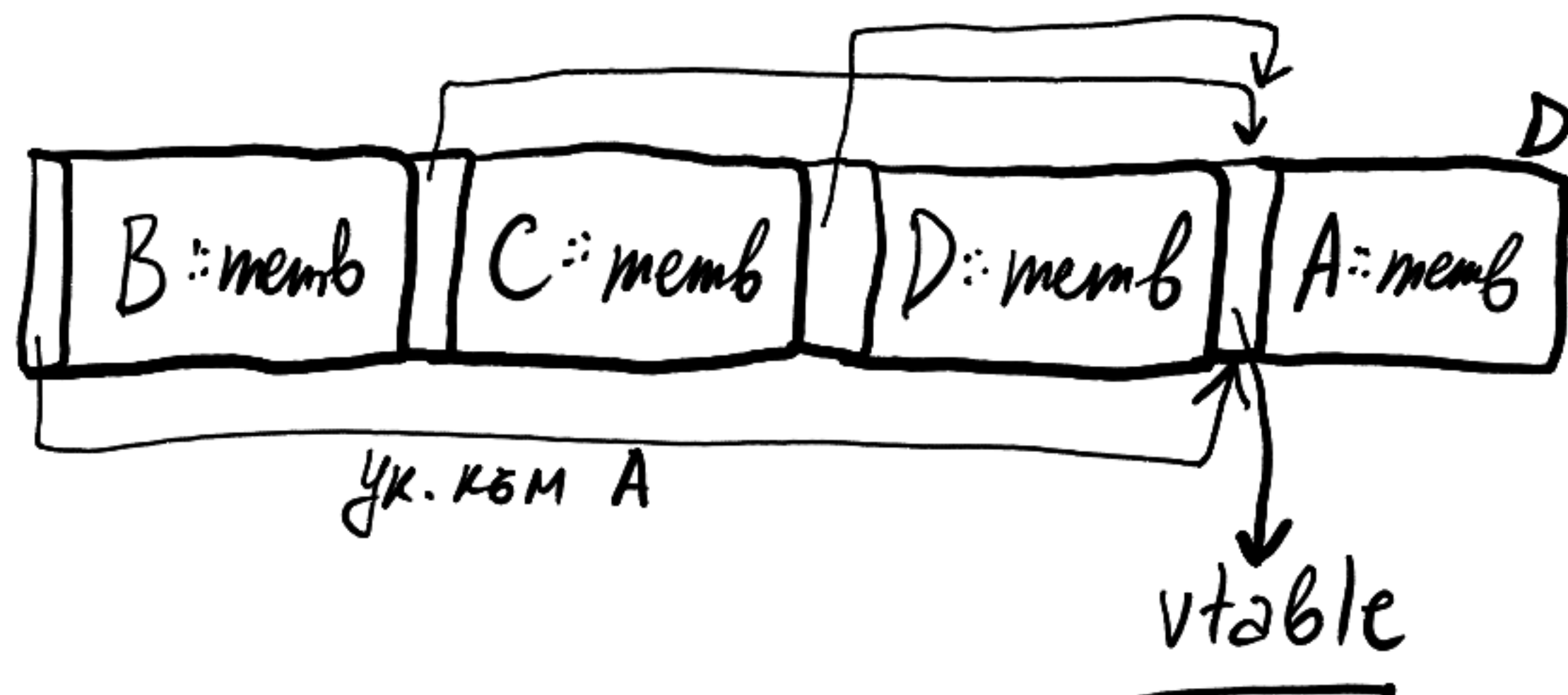
Ако искаме да няма дублиране на памет,  
използваме виртуално наследяване  
(X: virtual Y {})

Така V наследници на B и C (преки и  
непреки) отговарят за създаването на A

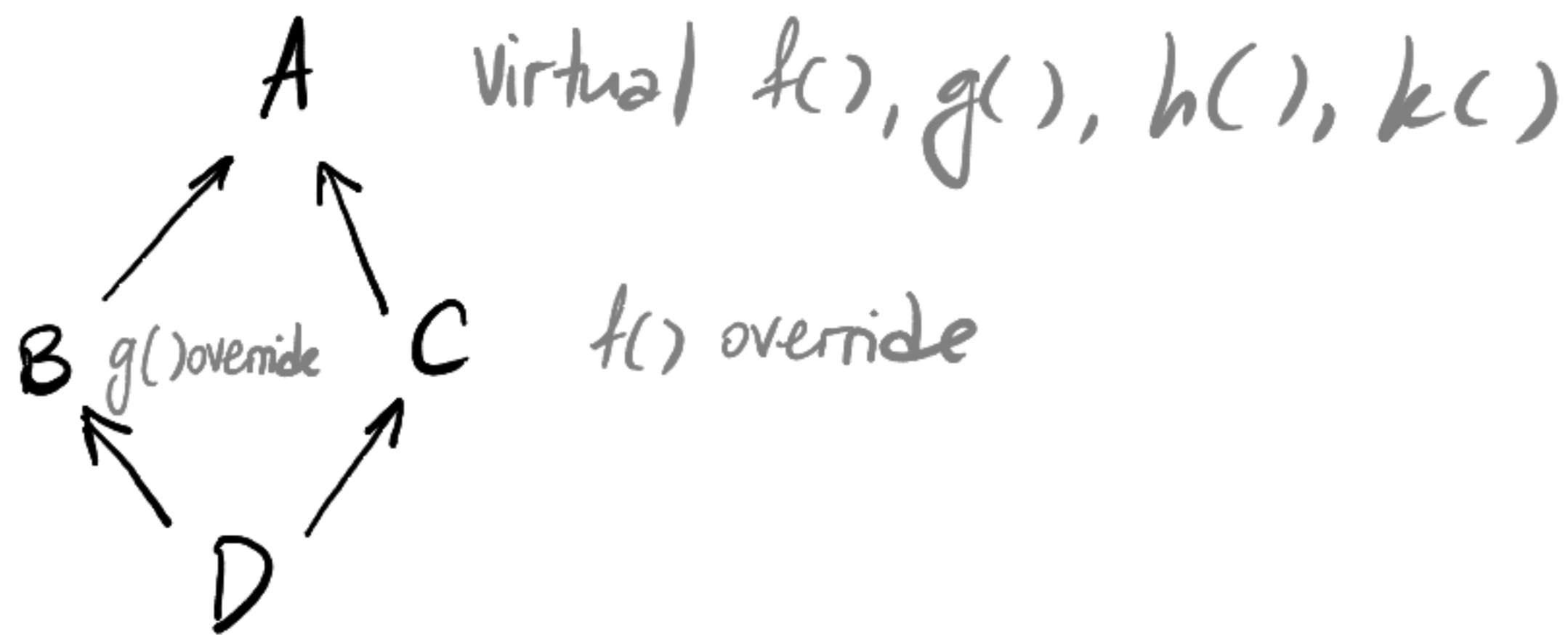
A: [A]

B: [B::memb] → [A]

C: [C::memb] → [A]







vtable of D:  
(от A)

$\Delta$	
A :: k	0
B :: g	$-\Delta(A)$
C :: f	$-\Delta(A) + \Delta(C)$
D :: h	$-\Delta(A)$

↑  
Тук не е

$\Delta(A) + \Delta(D)$ , защото там са само members of D,  
а  $-\Delta(A)$  сочи към целия обект D

## Колекции от обекти в полиморфна йерархия

- хетерогенен контейнер - клас, който съдържа колекция от указатели към абстрактен клас и се грижи за мениджърството на паметта
- копиране - чрез клониране - ф-я, която връща копие на себе си (дин. зададен обект от този тип)

```

Base* clone {
    return new A(*this);
}
  
```

- триене - виртуален деструктор
- разпознаване - чрез клас-данни / dynamic cast
- има конкретика само във factory