

# Структури

- Ако една структура съдържа обект от същия тип се получава безкрайна рекурсия (съжалявам, че не остана време за рекурсия)
- Едно коте чело книжка за едно коте, което чело книжка







# Структури

- ..... и така докато не свърши свободната памет и не избухнат фойерверки
- Може да съдържа поинтър към обект от същия тип
- Поинтърът не съдържа член данни и прочие, а само адрес, затова няма да се получи рекурсия както преди малко
- Не можете да зададете на такъв поинтър заделяне на динамична памет като параметър по подразбиране

```
4 #include "pch.h"
5 #include <iostream>
6 #include "Task3.h"
7
8 struct a
9 {
10     int * stupid = new int; //memory leak, but still won't cause compile error
11     a * arr = new a [5];
12 };
13
14
15 int main()
```

struct a  
the generated default constructor for "a" cannot be used in an initializer for its own data member

# Структури

- Съществува присвояване на стойностите на структури
- То може да се осъществява само между структури от един и същи тип
- Използва се оператор =
- Буквално се прехвърлят стойностите на всяка член данна 1 по 1

# Структури

example a = {1,2,3,4,5,6}, b = {8,5,3,1,6,7}; //да си представим, че има  
//само 6 член данни от тип int

a = b;

a	
member1	1
member2	2
member3	3
member4	4
member5	5
member6	6

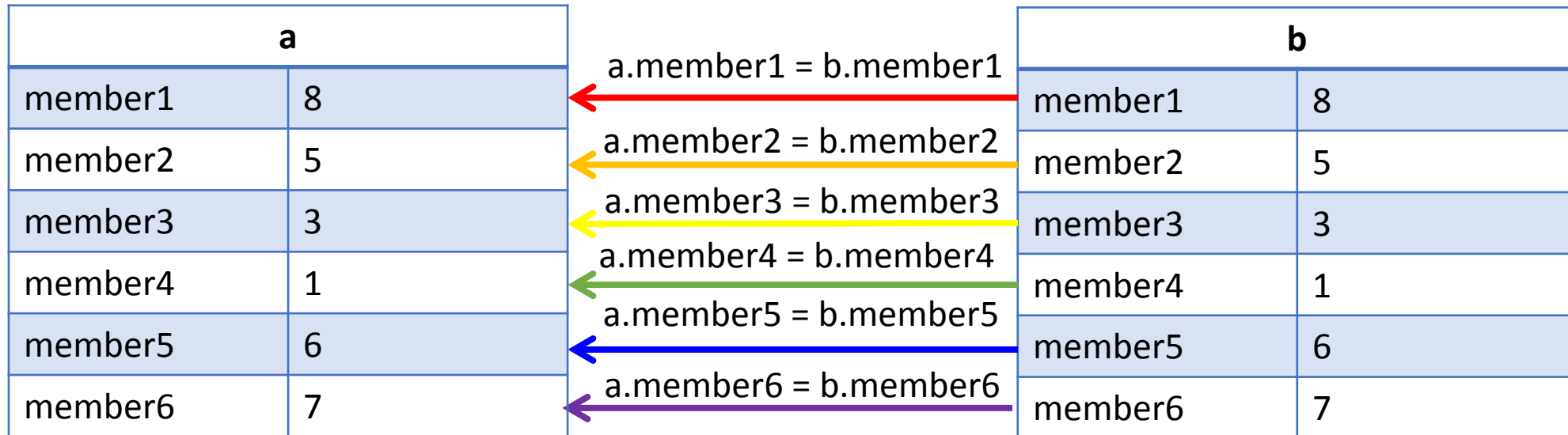
b	
member1	8
member2	5
member3	3
member4	1
member5	6
member6	7



# Структури

example a = {1,2,3,4,5, 6}, b = {8,5, 3, 1,6,7}; //да си представим, че има  
//само 6 член данни от тип int

a = b;



# Структури

- За структурите важат същите правила за подаване като параметър на функция и връщане като резултат както при примитивните данни
- При подаване като параметър, се създава нов локален обект, на който се присвоява стойността на подадения обект
- При връщане като резултат, се създава нов временен обект, на който се присвоява стойността на това, което връщаме
- Колко от вас видяха [Demo1](#) от миналата консултация