

Проблемът със споделянето на памет. Копиране и освобождаване

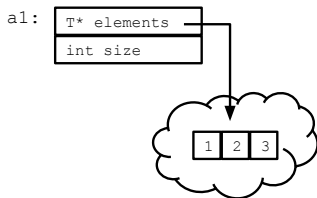
Калин Георгиев

22 март 2017 г.

Проблемът със споделянето на памет

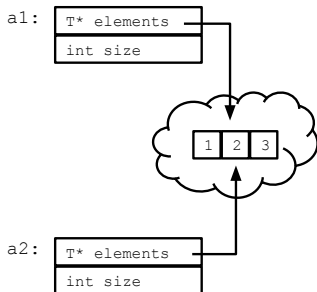
Обект и динамична памет

```
DynArray a1;  
//a1 <- 1,2,3
```



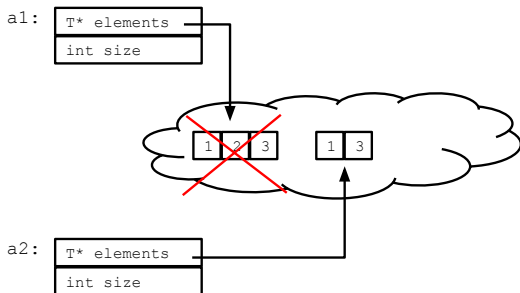
Инициализация чрез копиране

```
DynArray a1;  
//a1 <- 1,2,3  
DynArray a2 = a1; //a2(a1)
```



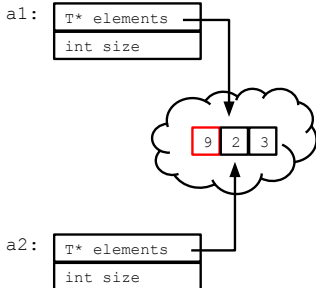
Операциите се отразяват на “общата памет”

```
DynArray a1;  
//a1 <- 1,2,3  
DynArray a2 = a1; //a2(a1)  
a2.remove (2);
```



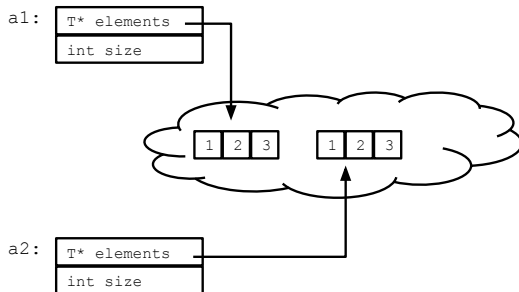
Операциите се отразяват на “общата памет”

```
DynArray a1;  
//a1 <- 1,2,3  
DynArray a2 = a1; //a2(a1)  
a2.elements[0] = 9;  
a1.print();
```



Решението е “истинско” копиране

```
DynArray a1;  
//a1 <- 1,2,3  
DynArray a2 = a1; //a2(a1)
```



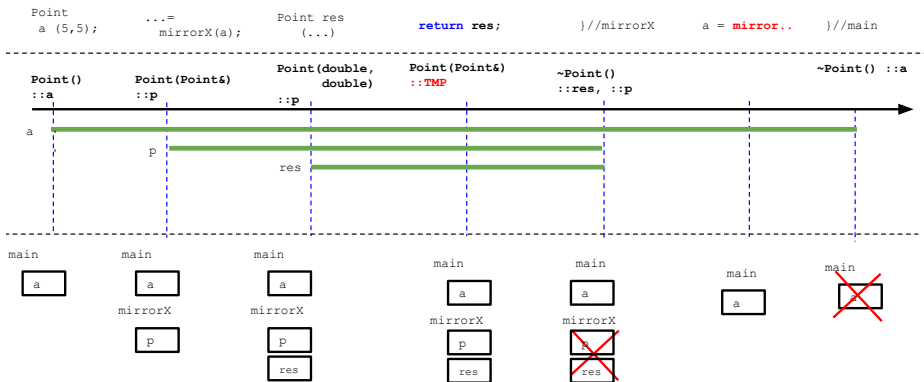
Случаи на копиране

Пример

```
class Point
{
public:
double x,y;

Point () {x=0;y=0;}
Point (double _x, double _y) {x=_x; y=_y;}
Point (Point &p) {x=p.x; y=p.y;}
Point (double _x) {x=y=_x};
};
```

Опростена схема

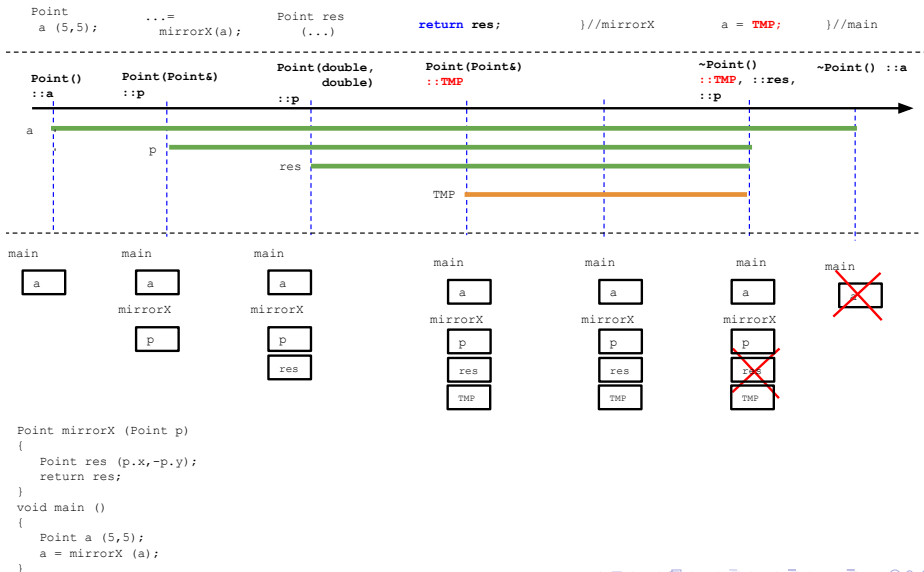


```

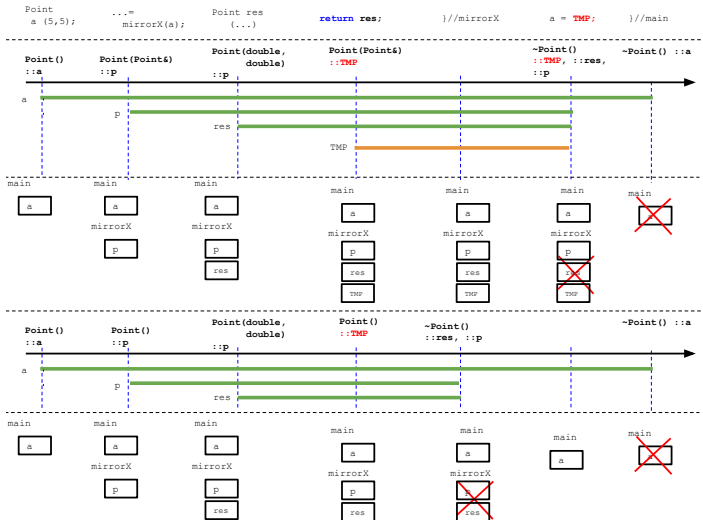
Point mirrorX (Point p)
{
    Point res (p.x,-p.y);
    return res;
}
void main ()
{
    Point a (5,5);
    a = mirrorX (a);
}

```

Пълна схема



Сравнение



Оператор за присвояване

Деструктор

Благодаря за вниманието!