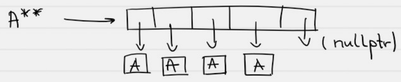
Тема 9

9 | Масиви от указатели към обекти. Move семантики - поглъщане, lvalue, rvalue, move конструктор/оп=, std::move. Пример за клас стринг с move семантика.

Масиви от указатели към обекти



Плюсове: + Не ни трябва def. кр на A
+ Позволяват се празни позиции (с nullptr)
+ Бързи swap-ове
+ resize не създава нови обекти

Въпреки това по-често се използват масиви от обекти (A*)
Заради locality - обектите са на съседни адреси
(итерацията и четенето стават по-бързо)

Move семантики

- откраднаме данните вместо да ги копираме
=> пестим време и памет

Типове данни:

lvalue - име на съществуваща променлива / функция
- извикване на ф-я, която връща реф.

rvalue = prvalue + xvalue
↘ expiring value

Литерали
Извикване на ф-я, връщаща копие

f(int a) - lvalue + rvalue
f(int& a) - lvalue
f(const int& a - lvalue + rvalue
f(int&& a) - rvalue && - rvalue ref

Когато подаваме обект по rvalue референцията, функцията откъсва подадената променлива/константа да не се използва повече след ф-ята, което значи, че тя свободно може да открадне данните ѝ

std::move - преобразува lvalue в rvalue, за да можем да крадем от вече създадени обекти
- декларираме, че от подадения lvalue нямаме да се използва след ф-ята

```
MoveFrom(A&& other){
    data = other.data;
    otherdata = nullptr;
}
```

move конструктор

```
A(A&& other) noexcept {
    moveFrom(std::move(other));
}
```

move оператор =

```
A& operator=(A&& other) noexcept {
    if(this != &other)
    {
        free();
        movefrom(std::move(other));
    }
    return *this;
}
```