

## Предфиниране на оператори

def Оператор-функция със специален синтаксис

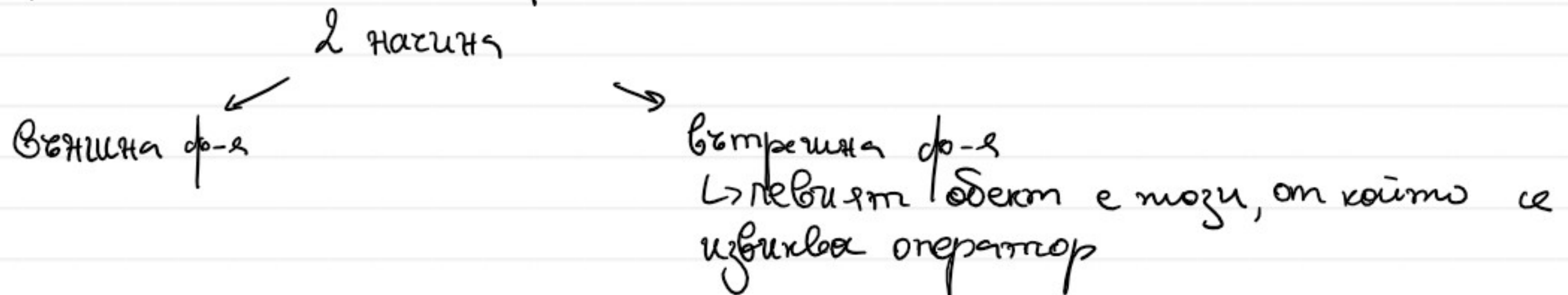
Видове: → унарни (1 аргумент) +, -, \*, &, ++, --  
→ бинарни (2 аргумента) +, -, \*, /  
→ тернарни (3 аргумента) ?:

• унарните оператори +, - не правят нищо, но се използват, когато динамично ще се определи коя операция ще се извършва

Характеристики:

- 1) Приоритет
- 2) Асоциативност - ляво (скобуваме отляво надясно), десно →  $((a \& b) \& c) \& d$   
→  $a \& (b \& (c \& d))$
- 3) Позиция спрямо аргументите си → инфиксен  
→ суфиксен  
→ префиксен

Предфиниране (за работа с определен тип)



Извикване

obj, obj2  
obj.operator \$(obj2) - алтернативен синтаксис  
obj + obj2  
this other

При предфиниране на оператори се спазват общоприети стандарти (без значение, се може да ги и накарваме да правят каквото и да е)  
Примери: 1) +, -, \* ... не променят аргументите си, а връщат нов обект  
2) +=, -=, -- променят първия аргумент и връщат некоего резултат  
3) ==, >, < ... не променят, връщат bool, сравняват

Особени случаи:

- оператори за поток - външни, за да бъде отляво потокът)
- оп. за индексация и деиндексация  
→ суфиксен - връща старата стойност на обекта  
→ префиксен - връща обекта (вече променен)

```

A operator++() { ... return *this; } // prefix
A operator++(int) { this->x++;
return A(x-1); } // suffix - тук стават дити
// нар-р само за да го разглеждаме
// от префиксния

```

- предфиниране 2 пъти оп. за индексация: за конст. и некоего достъп

Ограничения:

- 1) =, [], (), → - могат да бъдат само клас-ф-ии
- 2) !, ::, . - не могат да предфиниране
- 3) не могат да създават нови оператори
- 4) не могат да предф. - асоц. - приоритет
- 5) → трябва да връща - позиция  
обект - ун. (резултат)
- 6) ако предфиниране ||, & губи превратителното изчисление

## Приятелски класове и функции

- външни класове / ф-ии, които имат достъп до private член-данните

Синтаксис:

```

A {
private:
    int x;
public:
    friend void f();
    friend class B;
}

void f() {
    A obj;
    obj.x++;
}

class B {
    A obj;
    B() { obj.x = 7; }
}

```

x friend y;      z friend y;      - z не е приятелска за x!