Тема 7.

Предефиниране на оператори. Приятелски класове и функции.

Пример за реализация на комплексно число (7A) и Nvector (7B)

Оператор - функция със специален синтаксис

Видове оператори
- → унарни +, -, ++, --
- → бинарни +, -, /, *
- → тернарен ?:

Характеристики на операторите (operator overloading)
- → асоциативност - лява (+,-,*,/) и дясна (=)
- → приоритет
- → позиция спрямо аргумента
  - префиксни ++a
  - суфиксни a++
  - инфиксни

Предефиниране на оператори

- → позволява ни да дефинираме поведение на оператор
- → функции със специално име - ключова дума оператор и символът на респективния оператор
- → не могат да се предефинират асоциативността, приоритета и позицията спрямо аргумента

→ Не могат да се създават нови оператори
→ Някои оператори не могат да се предефинират
(?: , :: , .)

Има 2 начина за предефиниране:
- като външна ф-я

```
A operator $ (const A&, const A&)
```

+, -, *, /, както и оп. за поток << и >>
правим външни (потокът с левият аргумент)

- като вътрешна (променя левия аргумент)

```
A &
A operator $ (const A& other)
{
}
```

+=, -=, *=, /=, оп. =, оп ( ) предефинираме като член-функции, връщат референция към левия обект

Предефиниране на оператор ++:

```
A& operator ++ ( ); //prefix
A operator ++ (int n);
                    //suffix
```

Приятелски класове и функции

- външни класове/функции, които имат достъп до private и protected член-данни на текущия клас
- декларират се със запазената дума friend
- Не се наследяват
- Не са транзитивни