

# Конструктори

Трифон Трифонов

Обектно-ориентирано програмиране,  
спец. Информатика, 2019/20 г.

11–18 март 2019 г.

Тази презентация е достъпна под лиценза Creative Commons Признание-Некомерсиално-Споделяне на споделеното 4.0 Международен 

# Жизнен цикъл на обект

- За обекта се заделя памет и се свързва с неговото име
- Извиква се подходящ конструктор на обекта
- Работа с обекта (достъп до компоненти на обект, изпълняване на операции)
- Достига се края на областта на действие на обекта
- Извиква се деструкторът на обекта
- Заделената за обекта памет се освобождава

# Ролята на конструкторите

- Инициализират паметта за обекта
- Осигуряват, че преди да почне да се работи с обекта, той е във валидно състояние
- Позволяват предварително задаване на стойности на полетата

# Видове конструктори

- Обикновен конструктор с параметри
- Конструктор по подразбиране
- Конструктор с параметри по подразбиране
- Конструктор за копиране
- Системно генерирани конструктори
  - по подразбиране
  - за копиране
- Конструктор за преобразуване на тип

# Дефиниция на конструктор

```

<конструктор> ::=
    <име-на-клас> :: <име-на-клас> (<параметри>)
    [ : <член-данна> (<израз>) {, <член-данна> (<израз>) } ]
    { <тяло> }

```

## Пример:

```

Rational::Rational(int n, int d) : numer(n), denom(d) {
    if (denom == 0)
        cerr << "Нулев знаменател!";
}

```

Инициализиращият списък се изпълнява преди тялото на конструктора!

# Извикване на конструктори

`<описание на обект> ::=`  
    `<име-на-обект> [ = <израз> ] |`  
    `<име-на-обект> (<параметри> ) |`  
    `<име-на-обект> = <име-на-клас> (<параметри> )`

## Примери:

```
Rational r1, r2 = Rational(), r3(1, 2), r4 = Rational(3,4);  
Rational r5 = r1, r6(r2), r7 = Rational(r3);
```