

## Статично и динамично свързване

- Статично - изборът на ф-ята, която трябва да се извика, става по време на компилация
  - определя се от типа на ук-ля/референцията, от която се извиква ф-ята
- Динамично - изборът на ф-ята, която трябва да се извика, става по време на изпълнение на програмата

## Полиморфизъм

def 1 име на ф-я, много имплементации

Видове:
 

- Compile-time - function and operator overloading
- Run-time - динамично свързване

## Виртуални функции - чрез тях осъществяваме динамично свързване

- чрез ключовата дума virtual означаваме, че съответната функция може да се презаписва от наследниците на класа

- (при намяне на override се прави и проверка на синтаксиса)
- override - означаваме когато презаписваме виртуална ф-я
  - final → за виртуални ф-ии - забранява да се презаписва по-късно функцията
    - за класове - забранява наследяването на класа

Синтаксис:

```
class Base {
    virtual void f();
}
```

```
class Der : Base
{
    void f() override;
    ↳ Der::f() също е виртуална,
    защото виртуалността се
    наследява
```

По този начин можем да извикаме ф-ии от наследник през базов ук-л.

! При полиморфна иерархия деструкторът на Base винаги трябва да е виртуален, за да могат да се извикат и дестр. на наследниците

## Абстрактен клас

def чисто виртуални ф-ии - ф-я без имплементация, предназначена за презаписване от наследниците

```
class Base {
    virtual void f() = 0;
}
```

може да има тело, но може да се извика само в деструктор, на Der и в телото на презаписващата ф-я

- не може да се извика от външния свят

def Абстрактен клас - клас с поне една ч.в.ф. ⇒ предназначен за наследяване

→ не можем да създаваме обекти от такива класове  
→ наследник, който не презаписва ч.в.ф. също е абстрактен

## Виртуална таблица - "масив от ук-ли към ф-ии"

Таблица, в която пише коя ф-я трябва да се извика  
В всеки клас, съдържащ вирт. ф-ии има своя вирт. таблица

vpointer - всеки обект има виртуален ук-л, соещ към вирт. таблица на класа

- той е елемент-данни и заема памет