

11. Полиморфизъм. Статично и динамично свързване. Виртуални ф-ции. Ключови думи: `override`, `final`. Абстрактен клас. Виртуална таблица

Полиморфизъм

- 1 име много функционалности

I статично свързване (compile time polymorphism)

- избора на ф-ция става по време на компиляцията (предварително)
- определя се от типа на * или ref

1) function overloading

Пр: operators

$3 + 7$

"ABC" + "DCE"

Пр: обекти

class Base
f() 1..3

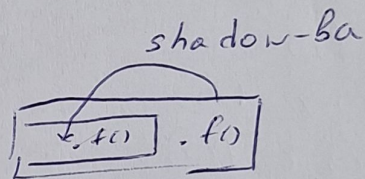
class Der: Base
f() 1..3

Der d;

d.f() // Der::f()

Base* b = &d

b.f() // Base::f()



II Динамично свързване (runtime polymorphism)

- избора на ф-ция става по време на изпълнение на програмата (runtime)

- чрез виртуални ф-ции

Деф. Виртуална ф-ция
член-ф-ция, която е дефинирана в базовия клас и предефинирана (`override`) в класа наследник

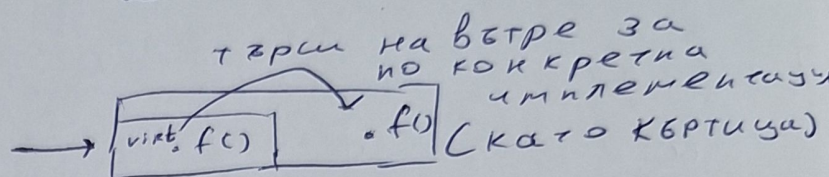
Пр:

```
Base  
virtual f()
```

```
Der: Base  
f()
```

```
Der d;  
d.f(); // Der::f()  
Base* b = &d;  
b.f(); // Der::f()
```

Диаг:



Override

- презаписване на виртуална ф-ция
- предупреждение за developers (показва грешки)

Пр:

```
Base  
virtual f()  
Base() { f(); }  
~Base() { f(); }  
g() { f(); }  
Der: Base  
f() override
```

```
Der d; // Base::f()  
d.g(); // Der::f()  
Base* ptr = &d;  
ptr->g(); // Der::f()  
③ // Base::f()
```

final

- последен ~~override~~ override на виртуал ф-ция
- не може да се override-ва натовъз по ъер

Пр: struct A
virtual f()

struct B
f() override

struct C
f() final

struct D
~~f() override~~

compiletime error

struct D: final: C { }

D - не може да бъде наследник

Абстрактен клас

- клас с поне 1 виртуална ф-ция
- от него не могат да се правят инстанции
- предказначен за наследяване

Чисто виртуална (поне virtual)

- наследника е длъжен да я предефинира
в противен случай и той ще се превърне
във абстрактен клас
- може да има и една имплементация

Пр:

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

```
Base::f() { ... }
```

```
class Der
    void f() override;
```

Виртуална таблица

- Таблица свързваща:

- типа на класа
- указатели към virt ф-ции
- преди извикване на virt ф-ция се търси във table
кой ф-ция да се извика

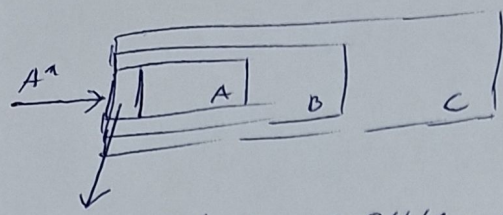
Пр:

A	virt f()	table A <div><div>A::f()</div><div>A::g()</div></div>
↑	virt g()	
B	f() override	table B <div><div>B::f()</div><div>A::g()</div></div>
↑		table C <div><div>B::f()</div><div>C::g()</div></div>
C	g() override	

концептуално

включване на table

Диаг:



vpointer - сочи към vtable на конкретния обект

vtable C

B::f()
C::g()

ptr → f();

1. Дереференциране ptr
2. Дереференциране vpointer
3. калирание ф-цията

трябва да си изясним до f(B^* this), но ptr е от тип A^*

4. кастване ptr към B^*
5. влизане във ф-цията