

Тема 11.

Статично и динамично свързване. Виртуални функции. Ключови думи - `override`, `final`.
Виртуални таблици. Полиморфизъм. Абстрактни класове.
Пример за фигури

Полиморфизъм - 1 име на функция, но много различни имплементации

- 1) compile time (function и оператор `overloading`)
- 2) runtime (дин. свързване)

Статично свързване - изборът на ф-я става по време на компиляция

- определя се от типа на указателя/реф., от който се извиква функцията

Динамично свързване - изборът на ф-я става по време на изпълнението на програмата

- постига се чрез виртуални функции
- виртуалната ф-я търси най-конкретната дефиниция

Виртуална функция

→ дефинира се в базовия клас, презаписва се в наследниците

→ гарантира, че ще се извика правилната ф-я, независимо от типа на ук. / реф.

→ не могат да бъдат `static`

- override → указва, че дадена ф-я презаписва ф-я от базовия клас. Не е задължителна, но може да спести грешки (ако няма такава ф-я в базовия клас няма да се компилира)

- final → указва, че дадена виртуална ф-я не може да се презаписва надолу по йерархията или че даден клас не може да се наследява

- чисто виртуална ф-я → виртуална ф-я, която няма имплементация
→ предназначена е да бъде презаписвана

- абстрактен клас → клас, който има поне 1 чисто виртуална ф-я и е предназначен за наследяване; не може да правим обекти от него; ако вирт. ф-я не се разпише от наследник и той става абстрактен; може да има тяло

Виртуални таблици

- Всеки клас, който има вирт. ф-ии, има своя виртуална таблица - в нея пише коя ф-я трябва да се извика
- Всеки обект има виртуален указател, който сочи към вирт. таблица на класа
- невиртуалните ф-ии не са в таблицата
- деструкторът е в таблицата