

Семинар 11 по ООП

Множествено наследяване, Диамантен проблем,
Чисто виртуални функции, Абстрактни класове



Множествено наследяване

- Както споменахме C++ предоставя възможност за множествено наследяване.
- Това е когато един клас наследява 2 или повече други
- Конструкторите на класовете се викат в реда на наследяването.
- Не се препоръчва, но на контролно ще бъдете force-нати да го използвате

Пример



Диамантен проблем

- Какво ще стане, ако наследим един базов клас в два други и после тези два наследим множествено в трети?
- Третият ще получи по две копия на базовия клас.
- Като пробваме да достъпим елемент на базовия клас, няма да знаем на кое копие искаме да го извикаме и ще върне грешка.

Пример



Решение

- Пак ще използваме добре познатото ни динамично свързване.
- Просто трябва двата класа да наследят родителския виртуално.

Пример



Въпроси?



Чисто виртуални функции

- Pure virtual (защото просто на английски звучи по-добре)
- Виртуална функция, за която не сме предоставили имплементация.
- Синтаксис:
 $\text{virtual } \langle \text{тип} \rangle \langle \text{име} \rangle (\langle \text{аргументи} \rangle) = 0$
- Използва се в базови класове, които само ще използваме за наследяване и за подтипов полиморфизъм.

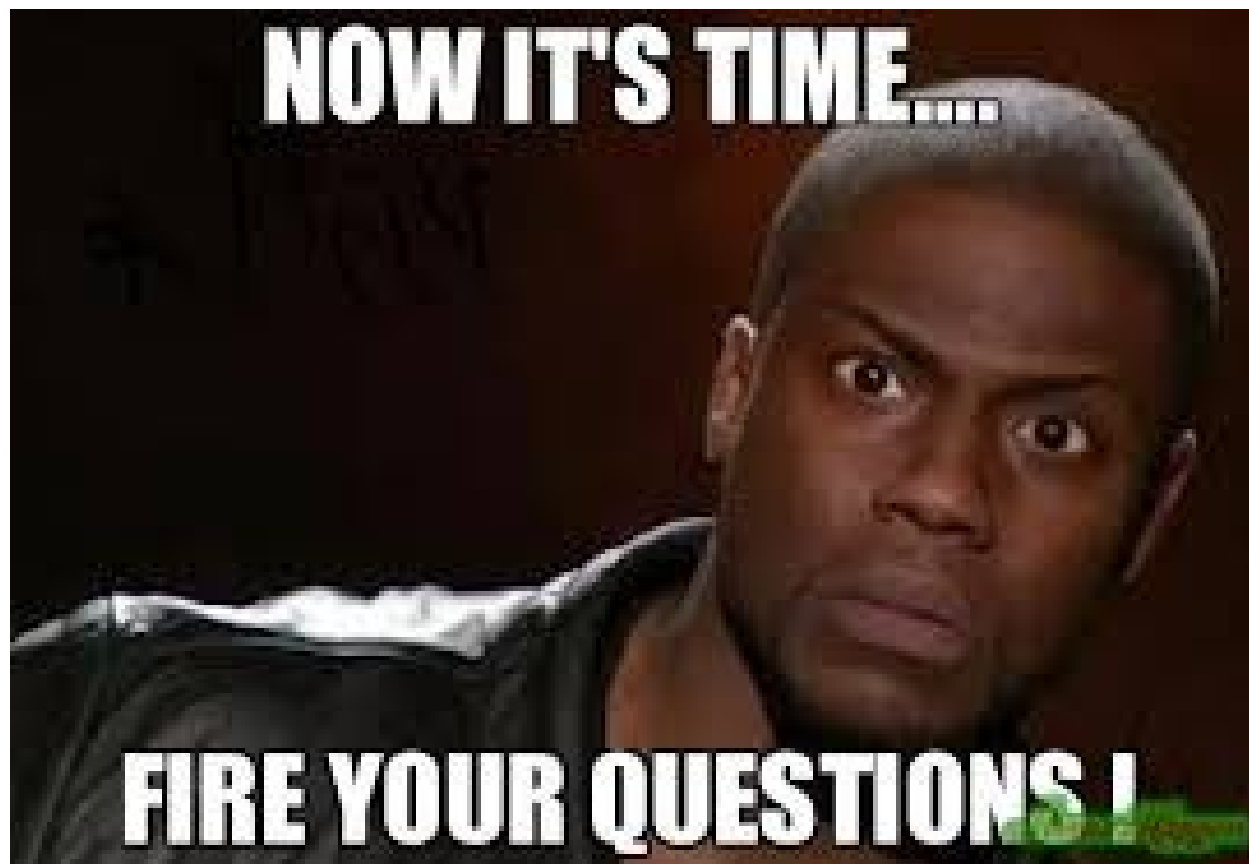
Абстрактен клас

- Клас в който има поне една чисто виртуална функция
- Не може да се инстанциира.
- Замества interface.
- Наследниците на абстрактен клас са абстрактни класове, освен ако не се дадат имплементации на всички чисто виртуални функции

Пример



Въпроси?



Задачи

- Да се създаде следната йерархия: Работник - Employee, който има име, възраст и номер, Шеф - Boss който е Employee и има функция getInfo, която приема работник и му извежда информацията, Работник в склад - StorageEmployee – който е Employee и има член данна storage, която описва в кой склад работи, StorageBoss, който е работник в склад и шеф. Нека всички тези да имат функция present, която извежда информация за тях.

Задача 2 (Контролно 2 миналата година)

- Една организация поддържа цялата си информация в документи и папки. Всеки документ има име, а съдържанието му представяме със символен низ. Папките също имат име, като освен това могат да съдържат документи, както и други папки, поставени в тях.
- [10 т.] Да се построи подходяща обектно-ориентирана йерархия, съдържаща класовете `TextDocument`, описващ документ и `Folder`, описващ папка.
- [25 т.] И двата класа да реализират метод `ListContents`, който извежда на стандартния изход информация за обекта. За документите това да е тяхното име, а за папките – списък с имената на всички съдържащи се в тях документи, включително документите, съдържащи се във вложените им папки.
- [25 т.] Освен `ListContents`, `Folder` да реализира операция `add` за добавяне на нов документ или папка.
- [25 т.] Да се реализират само необходимите конструктори, деструктори и оператори, включително виртуален конструктор за копиране, ако се налага.
- [10 т.] Да се построи в паметта следната папка, където елементите с пунктирани линии са текстови документи, а елементите с непрекъснати линии – вложени папки.
- [5 т.] За така построената папка да се изведе на екрана списък от всичките съдържащи се документи (редът няма значение):

Учредителен акт

Мария Иванова

Иванка Маринова

Производство

