

Objekt, třída

Třída

Třída: Elementární stavební prvek objektově orientovaného programování. Obsahuje předpis, podle kterého lze vytvořit instanci třídy - objekt. Obsahuje členské atributy a metody.

Zapouzdření: Třída zapouzdruje atributy a funkce jako "její". Na venek komunikuje pomocí rozhraní, nebo odhaluje některé atributy či metody pomocí přístupových práv, většinou

- Public - veřejné, k třídě či jejím public atributům a metodám lze přistupovat odkudkoliv
- Protected - třídě či jejím protected atributům a metodám lze přistupovat pouze z ní (this v případě objektu, statický přístup) nebo z odděděných tříd
- Private - třídě či jejím private atributům a metodám lze přistupovat pouze z ní

Některé jazyky podporují i další práva, např Internal v jazyce C#, který dovoluje přistupovat k třídě z její assembly (aplikace, dll knihovna,...)

Typy tříd:

- "Klasická" třída - dovoluje instancování na objekty, každá instance dostane přidělenou vlastní paměť
- Statická třída a členové - klíčové slovo static - Třída, která není instancovatelná, a její metody jsou přístupné pouze statickým způsobem, tedy Třída.Metoda() (nikoliv instanceTřidy.Metoda()). Klasická třída může obsahovat statické členy, které jsou pro všechny případné instance společné (například static parametr počet instancí, který se v konstruktoru zvýší). V jazycích, které statické třídy nepodporují se používá návrhový vzor singleton.
- Abstraktní třída - abstract - obsahuje abstraktní metody, není instancovatelná. Je nutné od ní oddědit a abstraktní metody doimplementovat (např abstraktní třída Čtečka s metodou Čti(), od které dědí FlashČtečka, DiskČtečka, CDROMČtečka....(To není zrovna dobrý příklad, jelikož čtení provádí OS, ale jako idea je to OK))
- Koncová třída (final v Javě, sealed v C#, v C++ není) - třída, kterou nelze dědit

Objekt

Objekt: Instance třídy, obraz třídy v paměti, referenční datový typ. Přebírá atributy a metody své třídy. Je nutné jej vytvořit pomocí operátoru new, případně uvolnit pomocí většinou delete (pokud není garbage collector).

Třída vs Struktura: Třída a struktura (struct, record,...) mohou mít společné vlastnosti. Rozdíly se projevují podle jazyka, většinou je však platný ten rozdíl, že členské prvky struktury jsou public zatímco členské prvky třídy jsou private. V C# je další rozdíl v uložení (třída v paměti, struktura na zásobníku). Java struktury nemá, používá se final class

Konstruktor a destruktork: Metoda, která inicializuje, respektive uvolňuje třídu do/z paměti. Většinou nastavuje atributy třídy na výchozí hodnoty.

Gettery a Settery: Veřejné metody, které nastavují privátní atributy objektu či třídy. Používáme je, abychom zamezili entitám využívající atributy třídy nastavit tyto atributy na nesmyslné hodnoty.