Objektově orientované programování (OOP)

Je programovací přístup založený na principu, že se naprosto vše dá popsat pomocí jednotlivých objektů.

Viz Programovací jazyky.

Základní pojmy

Objekt

Je základní jednotkou používanou v rámci OOP. Každý problém je pro OOP rozložitelný do objektů. Objekt v sobě uchovává vlastní data (atributy) a má i funkční části popisující jeho chování (metody).

Třída

Třída je prakticky „šablona objektu“. Popisuje, jaké bude mít objekt vlastnosti (atributy) a co dokáže dělat (metody).

Instance

Instance objektu je už běžící objekt (vytvořený podle vzoru třídy) s naplněnými atributy a s možností na něm volat implementované metody.

Atribut

Atributy určují jednotlivé vlastnosti objektu.

Mohou mít dva základní druhy:

- **Hodnotová vlastnost (spíše hodnotová proměnná)** – vlastnost vyjádřitelná jako číslo, znak nebo pravdivostní hodnota.
- **Objektová vlastnost** – další objekt, který původní objekt používá (například Člověk má jako vlastnost další objekty jako Srdce, Vlasy, Charakter…).

Metoda

Schopnost objektu provádět určitou činnost. Metody po svém provedení vracejí určenou hodnotu (například metoda Srdce *ziskejTlak()* by vracela současný krevní tlak) – může však vracet i hodnotu **void**, tedy nic (metoda *bij()* by například pouze provedla akci bití Srdce, nebylo by tedy třeba nic vracet).

Metoda může také přijímat potřebné argumenty (metoda Srdce *zmenRychlost(rychlost)* by nastavila novou rychlost bití Srdce podle zadané hodnoty argumentu rychlost).

Konstruktor

Speciální metoda konstruktoru, která je volána vždy při vytváření nové instance třídy (nevrací proto vůbec nic). Může zde docházet k inicializaci potřebných vlastností. Třída může mít i více konstruktorů podle počtu argumentů.

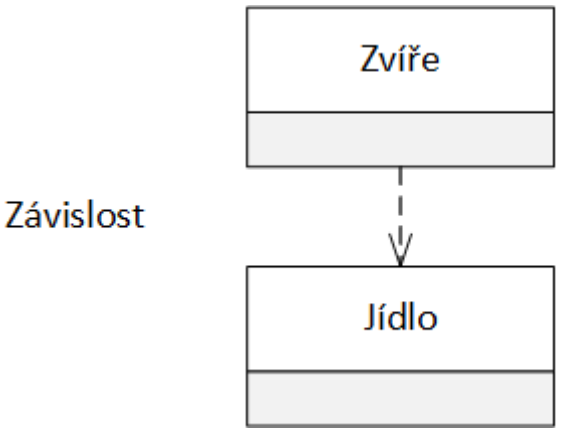
V některých objektových programovacích jazycích (C++) existuje i speciální metoda zvaná **destruktor**, která se používá, když už objekt není potřeba a má být zničen.

Vztahy mezi objekty

Závislost

Objekt, z kterého míří šipka, využívá ve svém kódu objekt do kterého šipka míří (například v argumentu metody nebo jako lokální proměnnou).

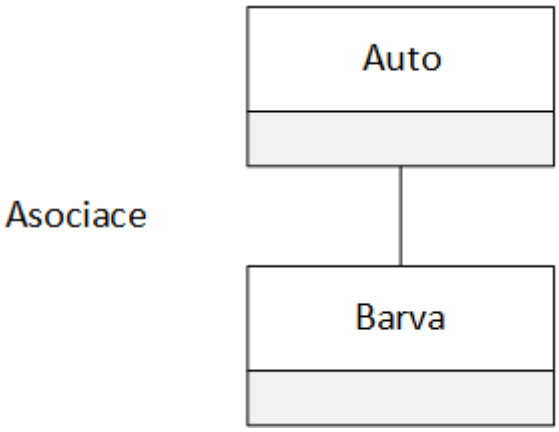
Př.: Object Zvíře obsahuje metodu nakrmit, která v rámci svého argumentu přijímá objekt typu Jídlo.



Asociace

Objekty jsou v užším vztahu přímo v kódu. Jeden obsahuje referenci na druhý (například jako datový typ atributu).

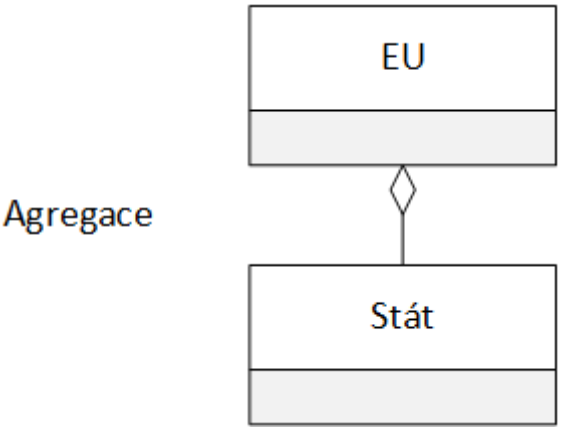
Př.: Objekt Auto obsahuje referenci na objekt Barva.



Agregace

Silnější vztah než asociace. Objekt na straně kosočtverce vlastní instance objektu na druhé straně. Při zániku prvního, ale nedochází k zániku druhých.

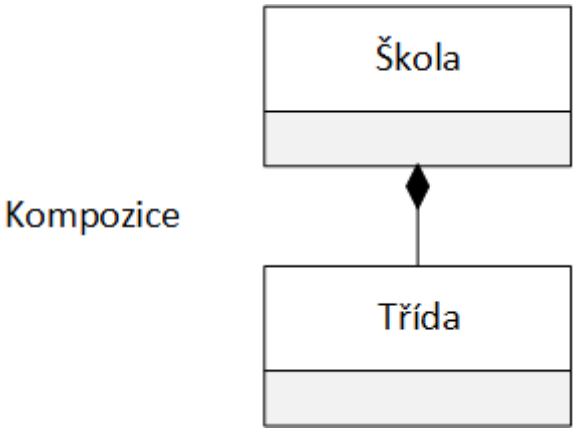
Př.: Objekt EU obsahuje referenci na objekty typu Stát, které jsou v rámci EU. Zanikne-li však EU, státy sice mohou válčit, ale pořád budou existují.



Kompozice

Silnější vztah než asociace. Objekt na straně kosočtverce vlastní instance objektu na druhé straně. Při zániku prvního zanikají i instance druhého.

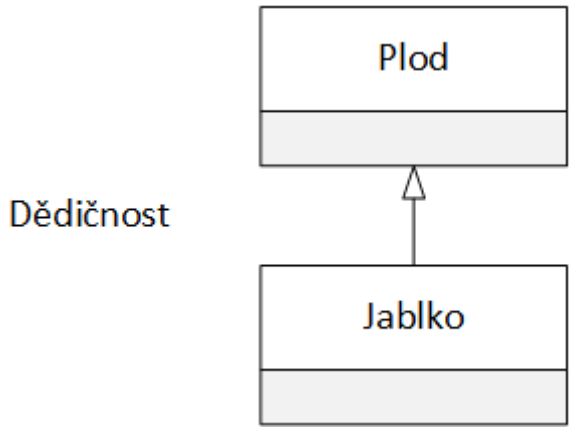
Př.: Objekt Škola obsahuje objekty typu Třída, které daná škola má. Ve chvíli, kdy Škola zaniká nemůžou dále existovat objekty typu Třída samostatně – zaniknou tedy společně se Školou.



Dědičnost (neboli specializace či generalizace)

Objekt, ze kterého míří šípka, je speciálním případem objektu druhého.

Př.: Objekt Jablko je speciálním případem objektu Plod.



Implementace

Speciální vztah mezi rozhraním a objektem, který ho implementuje. Význam je podobný jako u dědičnosti.

Př.: Objekt Auto implementuje (pohybové) metody rozhraní Pohybovatelný.