

# Objektově orientované programování (OOP)

Je programovací přístup založený na tom, že vše se dá popsat pomocí objektů.



Text hotový, chybí obrázky

## Základní pojmy

### Objekt

Je základní jednotkou používanou v rámci OOP. Každý problém je pro OOP rozložitelný do objektů. Objekt v sobě, jak data (v podobě atributů), tak má i funkční části popisující jeho chování (v podobě metod).

### Třída

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

### Instance

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

### Atribut

Jak už zde bylo několikrát zmíněno atribut je vlastností objektu. Vlastnosti může mít dva druhy:

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

### Metoda

Schopnost objektu provádět činnost. Může po svém provedení vracet nějakou hodnotu (Srdce má metodu ZískatTlak(), která vrací současný krevní tlak), ale nutně nemusí (Srdce má metodu Bij(), která nic nevrací).

Metoda může také přijímat argumenty (Objekt Malíř má metodu VymalujPokoj(Barva)).

## Konstruktor

Speciální metoda konstruktoru, která je volána vždy při vytváření instance dané třídy. Může zde docházet k inicializaci potřebných vlastností.

Pozn. V některých objektových programovacích jazycích existuje i speciální metoda zvaná destruktorka, 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 míří šipka (například v argumentu metody nebo jako lokální proměnnou).

Př. Objekt 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 kosočverce vlastní instance objektu na druhé straně. Při zániku prvního, ale nedochází k zániku druhých.

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

### Kompozice

Silnější vztah než asociace. Objekt na straně kosočverce 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.

## Dědičnost (nebo také specializace nebo generalizace)

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

Př. Objekt Jablko je speciálním případem objektu Plod.

## Implementace

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

Př. Objekt Auto implementuje metody rozhraní Pohybovatelný.

From:

<https://wiki.gml.cz/> - **GMLWiki**

Permanent link:

<https://wiki.gml.cz/doku.php/informatika:maturita:18a?rev=1424101872>

Last update: **16. 02. 2015, 16.51**

