

OOP (Objektově orientované programování)

OOP je způsob myšlení a návrhu softwaru, který se snaží modelovat reálný svět pomocí digitálních objektů. Tento přístup výrazně usnadňuje správu rozsáhlých programů, umožňuje znovupoužitelnost kódu a zvyšuje jeho přehlednost.

Mezi nejznámější jazyky využívající OOP patří **Java, C++, Python, C#, Ruby** nebo **Swift**.

Základní stavební kameny

OOP stojí na dvou klíčových entitách:

- 1. `[[class|Třída (Class)]]`**: Abstraktní vzor nebo šablona (např. výkres automobilu). Definuje, jaká data bude objekt mít a co bude umět.
- 2. `Objekt (Instance)`**: Konkrétní věc vytvořená podle třídy (např. konkrétní červené auto s SPZ 1A2 3456 stojící před domem).

Čtyři pilíře OOP

Aby byl systém považován za skutečně objektový, musí implementovat tyto čtyři principy:

1. Zapouzdření (Encapsulation)

Data (atributy) a operace nad nimi (metody) jsou uzavřeny v jednom celku - objektu. Vnitřní stav objektu je skrytý; vnější svět s ním komunikuje pouze skrze definované rozhraní. * **Výhoda:** Změna vnitřního fungování jedné třídy neovlivní zbytek programu.

2. Abstrakce (Abstraction)

Skrývání složitých detailů implementace a nabízení jednoduchého ovládání. Uživatel „auta“ ví, jak použít pedál (rozhraní), ale nemusí rozumět procesu vstřikování paliva v motoru. * **Výhoda:** Snižuje kognitivní zátěž programátora.

3. Dědičnost (Inheritance)

Umožňuje vytvářet nové třídy na základě již existujících. Nová třída (potomek) přebírá všechny vlastnosti a metody rodiče a může je rozšiřovat nebo upravovat. * **Výhoda:** Drasticky omezuje opakování stejného kódu (princip DRY – Don't Repeat Yourself).

4. Polymorfismus (Mnohotvárnost)

Schopnost různých objektů reagovat na stejný příkaz různým způsobem. Například třídy Čtverec a Kruh mohou mít obě metodu vykresli(), ale každá ji provede jinak. * **Výhoda:** Programátor může pracovat s obecnými typy, aniž by se staral o to, jaký konkrétní podtyp v danou chvíli používá.

Procedurální vs. Objektový přístup

Parametr	Procedurální (např. C)	Objektový (např. Java)
Hlavní jednotka	Funkce a postupy.	Objekty a data.
Přístup	Shora dolů (Top-down).	Zdola nahoru (Bottom-up).
Bezpečnost dat	Data jsou často volně přístupná.	Data jsou skrytá (zapouzďřená).
Údržba	U velkých projektů velmi obtížná.	Snadnější díky modularitě.

Kdy OOP použít?

OOP je ideální pro:

- **Složité systémy:** Simulace, GUI (grafická rozhraní), hry, webové aplikace.
- **Týmovou práci:** Různí programátoři mohou pracovat na různých třídách nezávisle na sobě.
- **Dlouhodobé projekty:** Kód se snáze rozšiřuje a opravuje.

Naopak pro velmi jednoduché skripty nebo výpočetně kritické nízkourovňové operace může být OOP zbytečně „těžkopádné“ kvůli režii spojené se správou objektů.

Související pojmy: Class, Instance, Atribut, Metoda, Konstruktor, Dědičnost, Polymorfismus, SOLID principy.

From:

<https://serviceit.cz/> - **IT ENCYKLOPEDIE**

Permanent link:

<https://serviceit.cz/doku.php?id=oop>

Last update: **2025/12/31 20:06**

