

Základy objektově orientovaného programování (OOP)

OOP je způsob nahlížení na programování, který se snaží modelovat digitální svět podobně jako ten reálný. Namísto psaní sekvencí příkazů (procedurální programování) vytváříme autonomní entity – **objekty** – které spolu komunikují.

1. Základní pojmy: Třída vs. Objekt

Pochopení rozdílu mezi třídou a objektem je klíčové:

- **Třída (Class):** Je to „vzor“ nebo „šablona“. Definuje, jaké vlastnosti a schopnosti bude mít výsledek. Příklad: Výkres k autu.
- **Objekt (Instance):** Je to konkrétní realizace třídy v paměti počítače. Příklad: Konkrétní modrá Škoda Octavia na parkovišti.

—

2. Čtyři pilíře OOP

Aby byl jazyk považován za plně objektový, musí podporovat tyto čtyři principy:

A) Zapouzdření (Encapsulation)

Data (atributy) a operace nad nimi (metody) jsou uzavřeny v jednom celku. Vnitřní stav objektu je skryt před vnějším světem a přístup k němu je možný pouze přes definované rozhraní (např. getter a setter metody).

- **Výhoda:** Zamezuje nechtěné modifikaci dat zvenčí.

B) Abstrakce (Abstraction)

Skrytí složitých implementačních detailů a vystavení pouze nezbytných funkcí. Uživatel objektu ví, **co** objekt dělá, ale nemusí vědět, **jak** to dělá uvnitř.

- **Příklad:** Řidič ovládá auto pomocí volantu a pedálů (rozhraní), aniž by musel rozumět spalování směsi ve válcích.

C) Dědičnost (Inheritance)

Umožňuje vytvářet nové třídy na základě již existujících. Potomní třída přebírá (dědí) vlastnosti a

metody rodičovské třídy a může je dále rozšiřovat nebo upravovat.

- **Příklad:** Třída `NákladníAuto`` dědí od obecné třídy `Vozidlo``.

D) Polymorfismus (Mnohotvárnost)

Schopnost různých objektů reagovat na stejné volání metody různým způsobem.

- **Příklad:** Máme metodu `zatrub()`. Objekt `Auto`` zatroubí „tůt“, zatímco objekt `Lod`` vydá hluboký tón, přestože oba voláme stejným příkazem.

—

3. Metody a Atributy

- **Atributy (Vlastnosti):** Data uložená v objektu (proměnné). Např. `barva``, `rychlost``, `hmotnost``.
- **Metody (Schopnosti):** Funkce, které objekt umí vykonat. Např. `jed()`, `zastav()`, `zmenBarvu()`.

—

4. Konstruktor a Destruktor

- **Konstruktor:** Speciální metoda, která se zavolá automaticky při vytvoření (instanciaci) objektu. Slouží k nastavení výchozích hodnot.
- **Destruktor:** Metoda volaná při zániku objektu, která uvolňuje prostředky (např. zavírá otevřené soubory).

—

5. Hlavní výhody OOP

- **Znovupoužitelnost:** Jednou napsanou třídu lze použít v mnoha projektech.
- **Udržovatelnost:** Program je rozdělen do logických modulů, což usnadňuje hledání chyb.
- **Škálovatelnost:** Snadnější správa rozsáhlých softwarových systémů.

Související články:

- [Přehled datových typů](#)
- [Datový typ Char](#)
- [Návrhové vzory](#)

Tagy: programming oop software-development abstraction inheritance polymorphism

From:

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

Permanent link:

<http://serviceit.cz/doku.php?id=it:sw:oop>

Last update: **2026/01/02 18:53**

