Objektově orientované programování

Osnova

- základní informace
    - koncepce
        - objekty
        - abstrakce
        - zapouzdření
        - kompozice
        - delegování
        - dědičnost
        - polymorfismus
- programovací jazyky
- teorie objektů
    - třída
    - dědičnost
    - metoda
    - atomizace

Základní informace

- = OOP
- programovací paradigma
- modelováno na reálném světě
    - snaha o 1:1 reprezentaci realita:počítač
- implementace různé mezi jazyky

Koncepce

- objekty
    - má třídu
        - je její instancí
- abstrakce
    - pokud potřebujeme opakovat nějakou činnost vytvoříme pro ni abstrakci (může být metoda)
    - abstrakce v kontextu tříd:
        - super třída nemusí implementovat nějakou metodu, přenechá to na podtřídách
- zapouzdření
    - každý objekt má rozhraní k přístupu
    - některé věci jsou prostě nedostupné z jiných objektů
- kompozice
    - objekt může obsahovat další objekty
- delegování
    - objekt může využívat služeb jiných objektů
        - provádí za něj operace
- dědičnost
    - třída může dědit od jiné třídy
    - stromová struktura 1:N
    - hloupý příklad:
        - Animal
            - Dog
                - Shepherd
                - Bulldog
            - Cat
            - Fish
- polymorfismus
    - objekt se chová podle toho jaké třídy je instancí
    - několik objektů může mít stejné rozhraní, pracuje na venek stejně, ale vnitřně se liší podle implementace
    - polymorfismus dědičnosti
        - tam kde očekáváme instanci nějaké třídy můžeme dosadit instanci podtřídy
        - void greetAnimal(Animal)
            - můžeme dosadit instanci
                - Dog, Cat, Fish
                - Ale i další nepřímé podtřídy: Shepherd, Bulldog

Programovací jazyky

- Java, C++, C#, PHP, Python, Rust, Go, …
- některé úplně OOP
    - Java, C#
    - vše musí být v nějaké třídě
- některé jen objekty umožňují, ale nemusíte je tvořit
    - Python, PHP

Teorie objektů

- třída
    - uspořádání informací do jedné entity
    - instance = objekt
        - atributy
        - operace: metody
            - zveřejnění → rozhraní
                - práce s objektem
        - zajímá nás jaké služby poskytuje, ne jak to provádí
            - zapouzdření
- dědičnost (viz koncepce)
    - překrytí
        - přepis metody → nová logika
    - podtřída
        - něco dědí, dítě
    - super třída
        - dědí se z ní, rodič
- metoda
    - rozhraní třídy
    - něco dělá
    - překrytí
    - přetížení
        - metody
            - rozhodnutí o tom jaká metoda bude volána dělá překladač
                - podle parametrů
                - můžu mít jedno jméno metody, ale různé parametry (datové typy i počet)
            - příklad:
                - prumer(int cislo1, int cislo2) a prumer(int cislo1, int cislo2, int cislo3)
                    - prumer(1, 1) vybere první
                    - prumer(1, 1, 1) vybere druhé
        - operátoru
            - C# – lze přetížit například operátor + a definovat tak co se stane, když sečtete dvě třídy
- atomizace (věc jako jednotka)
    - metoda řeší jeden konkrétní problém
    - objekt je určen také na jeden problém

Zdroje

- Dědičnost (objektově orientované programování). In: *Wikipedie* [online]. 2024 [cit. 29.03.2026]. Dostupné z: https://cs.wikipedia.org/w/index.php?title=D%C4%9Bdi%C4%8Dnost_(objektov%C4%9B_orientovan%C3%A9_programov%C3%A1n%C3%AD)&oldid=24142117
- Objektově orientované programování. In: *Wikipedie* [online]. 2025 [cit. 29.03.2026]. Dostupné z: https://cs.wikipedia.org/w/index.php?title=Objektov%C4%9B_orientovan%C3%A9_programov%C3%A1n%C3%AD&oldid=25348608
- Přetěžování. In: *Wikipedie* [online]. 2026 [cit. 29.03.2026]. Dostupné z: https://cs.wikipedia.org/w/index.php?title=P%C5%99et%C4%9B%C5%BEov%C3%A1n%C3%AD&oldid=25774178#P%C5%99et%C3%AD%C5%BEen%C3%AD_funkce