Class (Třída v OOP)

Třída definuje vlastnosti a chování, které budou mít všechny objekty z ní vytvořené. Zatímco třída je pouze teoretický popis (např. výkres domu), objekt je již existující entita (konkrétní postavený dům).

Struktura třídy

Každá třída se skládá ze dvou hlavních složek:

```
1. **Atributy (Vlastnosti / Data):** Proměnné, které uchovávají stav objektu (např. barva, jméno, velikost).
2. **Metody (Funkce / Chování):** Funkce, které definují, co objekt umí dělat (např. jet, mluvit, vypočítat cenu).
```

Vztah mezi třídou a objektem (Instanciace)

Proces vytvoření konkrétního objektu ze třídy se nazývá **instanciace**. Z jedné třídy můžeme vytvořit nekonečné množství objektů, z nichž každý může mít jiné hodnoty atributů.

Šablona (Třída: Auto)	Objekt A (Instance)	Objekt B (Instance)
Atribut: barva	červená	modrá
Atribut: značka	Škoda	BMW
Metoda: nastartuj()	(společné chování)	(společné chování)

Čtyři pilíře OOP spojené s třídami

Práce s třídami se opírá o čtyři základní principy:

1. Zapouzdření (Encapsulation)

Třída skrývá svá vnitřní data před vnějším světem. Přístup k nim je povolen pouze přes veřejné metody (např. pomocí tzv. getterů a setterů). Tím se zajišťuje, že nikdo omylem nezmění vnitřní stav objektu nevalidním způsobem.

2. Dědičnost (Inheritance)

Třídy mohou tvořit hierarchie. Nová třída (potomek) může převzít vlastnosti a metody existující třídy (rodič) a přidat k nim své vlastní. * *Příklad:* Třída `Pes` a `Kočka` dědí od obecné třídy `Zvíře`.

3. Polymorfismus (Mnohotvárnost)

Umožňuje různým třídám reagovat na stejné volání metody různým způsobem. * *Příklad:* Metoda `vydejZvuk()` u třídy `Pes` vyštěkne, zatímco u třídy `Kočka` zamňouká.

4. Abstrakce (Abstraction)

Třída umožňuje uživateli pracovat se složitým systémem pomocí jednoduchého rozhraní, aniž by musel rozumět vnitřní implementaci.

Speciální metody třídy

* **Konstruktor:** Speciální metoda, která se zavolá automaticky v okamžiku vytvoření objektu. Slouží k nastavení výchozích hodnot atributů. * **Destruktor:** Metoda, která se provede při zániku objektu (v moderních jazycích jako Java nebo Python se o toto často stará Garbage Collector).

Příklad (Pseudokód)

```
class Pes {
    // Atributy
    String jmeno;
    int vek;

    // Konstruktor
    Pes(String jmeno) {
        this.jmeno = jmeno;
    }

    // Metoda
    void stekej() {
        print(jmeno + " říká: Haf!");
    }
}
```

```
// Vytvoření objektu
Pes mujPes = new Pes("Alík");
mujPes.stekej(); // Výstup: Alík říká: Haf!
```

Související pojmy: Objekt, OOP, Instance, Dědičnost, Zapouzdření, Polymorfismus, Konstruktor.