# 12. Java - Objekty, atributy a metody

## Objekt × Třída

- Třída
  - Je abstraktní popis objektu, který popisuje jeho vlastnosti (atributy) a chování (metody)
  - Určuje, jaké vlastnosti a metody objekt bude mít, ale sama o sobě není objektem
  - Lze ji chápat jako návrh nebo plán pro objekty, které budou vytvořeny na základě této třídy
- Objekt
  - Je konkrétní instance třídy
  - Tzn. objekt je skutečným výskytem třídy, který má vlastnosti a chování, které jsou definovány touto třídou
  - Když vytvoříme objekt, musíme nejprve definovat jeho třídu, podle které se bude řídit
- Třída popisuje, jakým způsobem bude objekt vytvořen a jaké vlastnosti a chování bude mít,
- Objekt je konkrétním příkladem této třídy s konkrétními hodnotami atributů

*Copy-paste z otázky 11*

---

## Jazyková konstrukce (zápis) definice třídy objektů v Javě

- Definice třídy objektů v Javě je zapsána pomocí klíčového slova **class** následované názvem třídy a tělem třídy v závorkách
- Klíčové slovo **public** určuje přístupová práva k třídě, zde znamená, že třída je veřejná a může být použita v jiných částech programu
- Následuje název třídy s prvním písmenem velkým písmenem, aby se odlišil od názvů proměnných a metod
- Tělo třídy, které následuje v závorkách, obsahuje definice atributů, konstruktorů a metod, které jsou specifické pro danou třídu

### Konstruktor

- Definují se jako speciální metody tříd, které slouží k inicializaci objektů
- Každá třída může mít jeden nebo více konstruktorů
- Vytváří se pomocí klíčového slova public nebo private, následuje název třídy a v závorkách mohou být definovány parametry, které jsou předány konstruktoru při vytváření nového objektu

```
NazevTridy nazevObjektu = new NazevTridy(parametryKonstruktoru);
```

- Klíčové slovo **new** slouží k vytvoření nové instance třídy, tedy nového objektu
- Poté následuje název třídy, kterou chceme vytvořit, a v závorkách jsou předány parametry konstruktoru

## Výraz static

- Statické proměnné (také nazývané třídní proměnné) jsou definovány pomocí klíčového slova **static** a jsou sdíleny mezi všemi instancemi třídy
- To znamená, že všechny instance třídy mají přístup ke stejné statické proměnné a změna hodnoty této proměnné v jedné instanci ovlivní hodnotu této proměnné všude

```
public class Kruh {
    private int polomer;
    public static final double PI = 3.14159265359;

    public Kruh(int polomer) {
        this.polomer = polomer;
    }

    public double vypoctiObvod() {
        return 2 * PI * this.polomer;
    }
}
```

## Modifikátory přístupu k atributům a metodám

- **public**
  - Atributy a metody s tímto modifikátorem jsou přístupné z jakéhokoli místa, tedy jak zvnějšku třídy, tak z vnitřku třídy a dalších tříd
- **private**
  - Atributy a metody s tímto modifikátorem jsou přístupné pouze v rámci třídy, v které byly definovány, z vnějšku třídy nejsou přístupné
- **protected**
  - Atributy a metody s tímto modifikátorem jsou přístupné v rámci třídy, ale také z potomků této třídy
- **default** (bez modifikátoru)
  - Atributy a metody bez modifikátoru přístupu jsou přístupné v rámci stejného package

## Přetěžování konstruktorů

- Přetěžování konstruktorů je technika používaná v Javě pro vytvoření více konstruktorů pro třídu, každý s jinou sadou parametrů
- Tato technika umožňuje inicializaci objektů třídy různými způsoby v závislosti na potřebách programátora
- Pro přetížení konstruktorů můžete definovat více konstruktorů pro třídu, každý s jiným počtem a/nebo typem parametrů
- Signatura konstruktoru musí být jedinečná, což znamená, že každý konstruktor musí mít jiný počet nebo typ parametrů
- Když se vytváří nový objekt třídy, volá se konstruktor, který má odpovídající počet a typ parametrů
- Pokud neexistuje konstruktor s odpovídajícími parametry, dojde k chybě kompilace
- Pokud nebyly specifikovány žádné konstruktory, kompilátor automaticky vytvoří konstruktor bez parametrů

## Prakticky

- Vytvoření definice třídy, přidání nějaké jednoduché metody pro práci s atributy (zkusit getery a setery)