1. Java - Principy objektově orientovaného programování

# Objektově orientované programování a jeho účel (důvody vzniku, přínosy)

* Je programovací paradigma, které se zaměřuje na organizaci kódu kolem datových struktur a funkcí, nazývaných objekty
* Každý objekt má svůj vlastní stav, chování a identitu, což umožňuje programátorům vytvářet komplexní a rozsáhlé systémy
* Důvody vzniku OOP se liší podle zdroje, ale některé z hlavních důvodů zahrnují potřebu větší modularity, efektivnější znovupoužitelnosti kódu a lepšího zvládání komplexity softwarových systémů.
* Výhody – větší flexibilita, schopnost zacházet s velkými objemy dat, snadnější údržba a rozšíření kódu a zlepšená ochrana dat

# Vztah mezi objektem a třídou

* 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

# Atributy třídy

* Jsou proměnné, které jsou definovány v rámci třídy a jsou sdíleny mezi všemi instancemi této třídy
* Atributy třídy jsou vlastnosti, které mají společné všechny objekty, které jsou vytvořeny z této třídy

# Základní principy objektově orientovaného programování

* Zapouzdření
  + Umožňuje skrýt vnitřní stav a chování objektu před ostatními objekty v programu
  + Zajišťuje, že objekt může být modifikován pouze přes jeho veřejné rozhraní, které je obvykle tvořeno metodami objektu
* Dědičnost
  + Umožňuje vytvářet nové třídy na základě již existujících tříd, které jsou označeny jako "rodičovské třídy" nebo "nadřazené třídy"
  + Umožňuje vytvářet hierarchie tříd, které mohou sdílet určité vlastnosti, chování a metody
* Polymorfismus
  + Umožňuje objektu používat stejné jméno metody, ale s různými implementacemi v různých třídách
  + Umožňuje efektivní a elegantní řešení problémů, které se týkají více objektů

# Prakticky

* Popsat na kódu principy tři OOP