

Programování

OOP rekapitulace





Objektové modelování v OOP

- ▶ = abstrakce reálných objektů pro potřebu implementace
- ▶ Míra abstrakce se liší od použití
- ▶ Princip modelování objektu:
 - ▶ Abstrakce
 - ▶ Formalizace
 - ▶ Jednoznačnost
 - ▶ Zamezení redundancí



Koncepce OOP - dědičnost

- Vytváření hierarchie objektů
- Odvozená třída dědí atributy a metody třídy nadřazené
- V C# je nejvyšší třídou, ze kterého je každá třída odvozena, třída **Object**
- Princip dědičnosti je využit i v databázových modelech
- Na předchůdce třídy přistupujeme pomocí klíčového slova **base**

Koncepce OOP - zapouzdření

- ▶ Vytvářená třída může mít libovolné množství členů
- ▶ Při správném použití konceptu zapouzdření jsou data skryta uvnitř třídy
- ▶ K hodnotám přistupujeme skrze metody
- ▶ Viditelnosti:
 - ▶ Public
 - ▶ Private – používáme pro skrytí před vnějším přístupem
 - ▶ Protected

Koncepce OOP - polymorfismus

- Koncept umožňující, aby akce uskutečněné nad objektem měly stejný/obdobný výsledek, ale realizace se liší
- Realizace (implementace) je odlišná právě od typu objektu
- **Virtuální** metody
 - Označení metod, kterým umožníme k reimplementaci v třídě odvozené
 - V případě, že třída tuto metodu reimplementovanu nemá, použije se řešení z třídy nadřazené
 - Přepsanou virtuální funkci označujeme klíčovým slovem **override**
 - V těchto funkcích můžeme pomocí **base** volat implementaci předešlou
- Override a virtual nelze použít u privátních funkcí třídy



Abstraktní metody a třídy

- Klíčové slovo **abstract**
- Abstraktní metody neobsahují implementaci, ale pouze hlavičku volání dané funkce
- Abstraktní metody jsou automaticky virtuální
- Třída obsahující abstraktní metodu = abstraktní třída
- Z abstraktní třídy nelze vytvořit novou instanci (objekt)
- Jako abstraktní mohou být uvedeny i události a vlastnosti třídy



Rozhraní

- Rozhraní (interface) definuje popis funkcností třídy, která je z daného rozhraní odvozena
- Obsahuje seznam členů a položek, které se odvozená třída (i struktura) zavazuje implementovat
- Jedno rozhraní může dědit od jednoho či více rozhraní
- Pozor, třída může dědit pouze z jedné třídy!
- Jedna třída může implementovat jedno či více rozhraní
- Všechny členy jsou automaticky **public**
- Z rozhraní rovněž nelze vytvořit novou instanci