# Objektově orientované programování 1

Paradigmata = programovací styly

## Strukturované

* Využívá podmínek, cyklů, podprogramů, skoky

## Objektové

* Objekty obsahující data
* Obsahuje praktiky ze strukturovaného
* Více souborů

## Objekt

* Instance
* Místo v paměti nastavené třídou nebo strukturou = vzor
* Program může tvořit více objektů stejné třídy nebo struktury
* Uložen v proměnné nebo kolekci
* Třída Object – nevyšší základní třída

## Základní objektové vlastnosti

### Dědičnost

* Tvoření nových datových struktur na základě starých
* Dědící třída dědí, rozšiřuje, modifikuje chování rodičovské třídy (člověk, student, pracující)
* Single inheritance only v C#
* Konstruktory a destruktory se nedědí – každá třída vlastní :base()
* Znovupoužitelnost kódu a méně redundance

### Zapouzdření

* Zabalení dat a metod a restrikce k nim
* Pro skrytí dat ve třídě
* Omezuje chyby – použití špatných metod atd…
* **Private** – ke členům třídy mají pouze třídy, které dědí nebo jsou nested, vloženy
* **Private protected –** pouze ty, co dědí a jsou ve stejném sestavení
* **Protected** – pouze ty, co dědí
* **Internal** – všechny ve stejném sestavení
* **protected internal**  – všechny třídy ze stejného dll a ty, co dědí
* **public** – všechny třídy

### Polymorfismus

* metody, které mají všichni potomci definované stejnou hlavičkou
* jiná implementace stejných metod
* virtual + override

## Třídy

* Referenční datový typ
* Vzor pro tvorbu objektů

### Abstraktní třída

* Abstract, může používat i metoda, vlastnost, indexer, událost
* Pouze jako základ pro ostatní třídy
* Nelze vytvořit její instanci
* Abstraktní členy pouze v abstraktních třídách
* Pouze hlavičky, tělo v ostatních dědících třídách
* + override (abstract je automaticky virtual)

### Statická třída

* Nelze z ní dědic ani vytvářet instance
* Všechny členy také statické
* Přistupujeme přes jméno třídy

### Zapečetěná sealed třída

* Z třídy nelze dědit
* Modifikátor lze použít i na přepsané override metody, aby nešly přepisovat v ostatních potomcích

### Partial třída

* Částečná třída
* Definice tříd a struktur lze rozdělit do více souborů
* Všechny části spojeny při kompilaci

### Privátní konstruktor

* Kromě nested potomků nemohou ostatní třídy vytvářet instance
* Např, když máme metodu, která nám vyhodí instanci

## Rozhraní interface

* Informace, které má třída obsahovat
* Definuje atributy, vlastnosti, metody třídy
* Všechny členy automaticky veřejné
* Třída implementující musí mít všechny členy
* Nelze vytvořit instanci
* Lze dědit pouze z jedné třídy
* Lze dědit z mnoha rozhraní