**OOP – Object-oriented programming**

* Programování založeno na vytváření objektů
* První program který umožňoval OOP byla Simula
* Některé programovací jazyky (Ruby, Scala) berou vše jako objekt
* Vhodné pro složité programy

**Třídy**

* Samostatná datová jednotka
* Obsahuje metody a atributy (vlastnosti) třídy
* Atributy a metody se přidělují každému objektu v této třídě
* A black background with blue and white text

  Description automatically generatedA black screen with white text

  Description automatically generatedTvorba pomocí keywordu class
* Metody a atributy v třídě přísluší pouze třídě a nejsou přístupné odjinud z programu
  + Zpřístupnit je lze pomocí keywordu global (v C++ public)
* Třída nemůže být prázdná, musím použít pass statement

**Objekty**

* Jednotliví zástupci třídy
* Své vlastnosti přebírají od třídy
* A black and white image of a math equation

  Description automatically generatedTvorba

**Build-in funkce**

* **\_\_init\_\_():** - používá se k přidělení hodnot a provedení operací při vytvoření objektu
  + Používá parametr self (musí být na prvním místě parametrů při volání funkce)
  + Self poukazuje na určitý objekt
* **\_\_str\_\_():** - kontroluje co se vrací když je objekt reprezentován jako string

**Metody**

* Funkce příslušící danému objektu a třídě
* A black background with orange and white text

  Description automatically generatedTvorba
* Obsahují parametr self, který má instance definované již z funkce \_\_init\_\_()

**Vlastnosti tříd**

* **Inheritace** (dědičnost) – třída může dědit vlastnosti jiné třídy
  + Při vytváření třídy píši do závorek třídu, ze které nová dědí
* **Encapsulation** (zapouzdření) – třídy se navzájem neovlivňují
* **Abstraction** (abstrakce) – proměnné a metody definované v třídě nejsou globální
* **Polymorphism** (polymorfismus) – různé třídy můžou mít metodu se stejným názvem

**Měnění tříd**

* Úprava atributů
* Mazání atributů
* Mazání tříd