



PODSTAWY PROGRAMOWANIA W PYTHON

PO 11 ZAJĘCIACH:

Do wypełnienia ankieta:

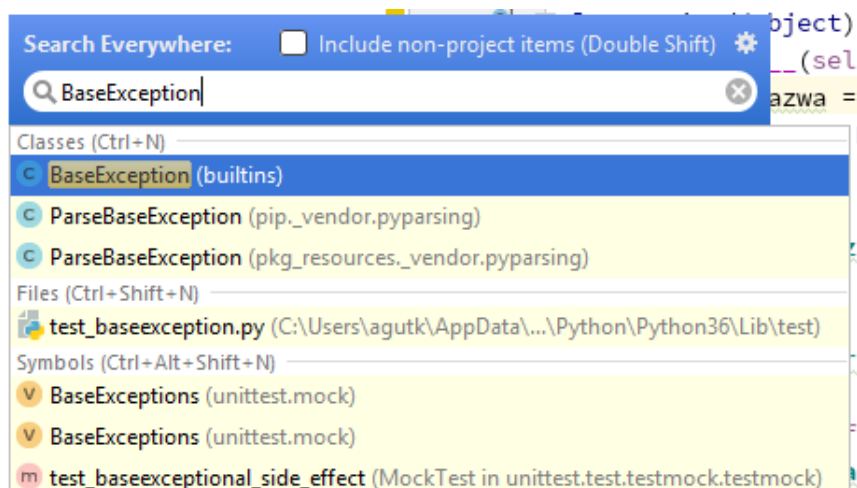
<https://goo.gl/forms/NExRAN3XuRRTNlcB3>

1. Omawiane zagadnienia:

- a. **klasy** – kontynuacja – definiowane klas, używanie instancji
- b. **klasy** – przeciążanie (overriding) metod i atrybutów specjalnych:
operatory: +, -, ==, <, >
zachowanie metod: print, len itp.
Metody specjalne – dunder (double **underscore**) podwójny podkreślnik przed i po nazwie: `__init__` `__str__` `__add__` `__lt__` `__gt__` itp.
- c. **paradygmaty programowania obiektowego**: abstrakcja, enkapsulacja, dziedziczenie, polimorfizm
- d. **klasy – dziedziczenie**:
 - i. klasa bazowa (rodzic, nadrzędna) – używa w sygnaturze `object`,
 - ii. klasy dziedziczące (dziecko, podklasa) – w sygnaturze używamy nazwę klasy z której dziedziczymy
 - iii. klasa dziedzicząca może być rodzicem dla innej klasy
 - iv. z jednej klasy może dziedziczyć wiele klas
 - v. jedna klasa może dziedziczyć z wielu klas
 - vi. dziedziczenie odbywa się z góry na dół
 - vii. podklasa korzysta z implementacji istniejących w klasie nadrzędnej, może też zdefiniować własną implementację

2. Kod Python też jest zdefiniowany w ten sam sposób jak tworzymy własne obiekty. Mielśmy styczność z klasami, które miały wspólne atrybuty, a jednak się różniły – były to wyjątki. Zobacz jak to wygląda w kodzie Python – W PyCharm naciśnij dwa razy klawisz Shift, pojawi się okienko wyszukiwania, wpisz w nim `BaseException` i

naciśnij enter – PyCharm otworzy i przeniesie Cię w odpowiednie miejsce modułu builtins. Zobacz w jaki sposób wyjątki dziedziczą z klasy BaseException.



Możesz też poszukać innych klas dziedziczących (np. NamedTuple)