



# PODSTAWY PROGRAMOWANIA W PYTHON

PO 9 I 10 ZAJĘCIACH:

## ANKIETA DO WYPEŁNIENIA

<https://goo.gl/forms/kuNXXeMb1Gdn8MID2>

### 1. Omawiane zagadnienia:

#### a. definiowanie klas

- i. słówko kluczowe **class**
- ii. definiowanie konstruktora **\_\_init\_\_**
- iii. znaczenie słówka **self** – oznacza odniesienie do konkretnej instancji obiektu
- iv. definiowanie pól/atributów/zmiennych instancji: **self.zmienna**
- v. definiowanie metod klas – **def funkcja(self, ...):**

#### b. używanie klas

- i. tworzenie instancji klas:  
**instancja = MojaKlasa()**
- ii. odwoływanie się do atrybutów/zmiennych instancji  
**instancja.atribut**
- iii. wywoływanie metod instancji:  
**instancja.metoda()**

#### c. Używanie instancji obiektów jako atrybutów wewnątrz innych obiektów

```
auto = Samochod()
maly_silnik = Silnik()
auto.silnik = maly_silnik
auto.silnik.moc
```

#### d. klasy – przeciążanie (overriding) metod i atrybutów specjalnych:

operatory: +, -, ==, <, >

zachowanie metod: print, len itp.

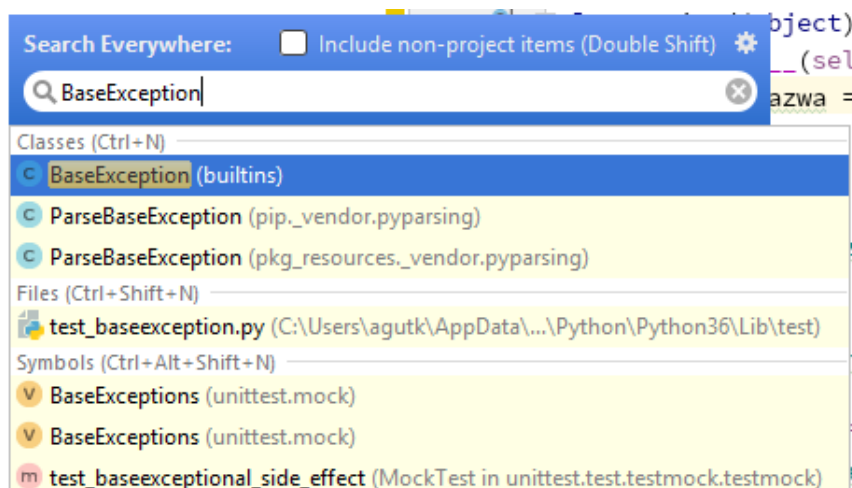
Metody specjalne – dunder (double underscore) podwójny podkreślnik przed i po nazwie: `__init__` `__str__` `__add__` `__lt__` `__gt__` itp.

e. **paradygmaty programowania obiektowego:** abstrakcja, enkapsulacja, dziedziczenie, polimorfizm

f. **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. Mieliśmy 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`)

3. Zadania do domu (na weekend):

- a. stworzyć jak najwięcej klas, które odwzorowują otaczającą nas rzeczywistość, np.: telewizor, grzejnik, piekarnik, czajnik, drzewo, kwiat, osoba, napój, komputer itp. itd.

- b. tworzyć instancje tych klas – sprawdzać jak się zachowują, nauczyć się wydobywać z nich informacje, korzystać z metod
- c. jeśli czujemy się pewniej, to stwórzmy klasy, których atrybutami są inne klasy
- d. możemy zastanowić się, w jaki sposób nasza baza może wykorzystać obiekty i zaimplementować bazę z klasami