**Zajęcia 05\_zadania\_na zajęcia**

**Zadanie 1.**

Odnosząc się do zadania z zajęć ostatnich (dotyczące implementacji Stworów)

Zaimplementuj klasę bazową (dowolną). Następnie bazując na modelu dziedziczenia hierarchicznego utwórz 3-4 klasy pochodne po klasie bazowej oraz kolejne 2-3 klasy potomne (dziedziczące w drugiej linii po 3-4 klasach pochodnych).

Z wykorzystaniem odpowiedniej metody sprawdź czy wybrane klasy pochodne dziedziczą po klasie bazowej.

Zaimplementuj metody dla klasy bazowej. Wywołaj metody dla obiektów różnych klas (pochodnych).

W kolejnym kroku zaimplementuj metodę o tożsamej nazwie w klasie pochodnej jak w bazowej. Wywołaj tę metodę na obiektach różnych klas. Zaobserwuj wynik. Jeżeli nie jest on do końca jasny wywołaj na danym obiekcie metodę help(object) albo metodę \_\_mro\_\_ na klasie.

Zaimplementuj w klasach atrybut o tej samej nazwie. Jak się odwołać do atrybutu aby uzyskać wartość zadeklarowaną w danej klasie.

Zaimplementuj klasę bazową, która w konstruktorze będzie przyjmować dwa argumenty. Następnie zaimplementuj klasę pochodną, która dziedziczy po klasie bazowej, ta klasa z kolei w konstruktorze przyjmuje trzy argumenty (dwa z nich są identyczne jak w klasie bazowej).

W dalszej części optymalizuj swój kod – wykorzystaj metodę super() zarówno dla definiowanych atrybutów w klasach pochodnych jak również metod.

Podobnie zaimplementuj metodę w klasie bazowej i wywołaj ją dla obiektów będących instancjami obu klas.

Zwracając wyniki w output np. z wykorzystaniem funkcji print zastosuj metody magiczne tj. \_\_class\_\_ oraz \_\_name\_\_ (zastanów się co one zwracają) – aby wyświetlane wyniki miały sens.

**Zadanie 2.**

Zaimplementuj dowolną klasę bazową, która w konstruktorze będzie przyjmować dwa argumenty. Następnie zaimplementuj klasy pochodne, która dziedziczy po klasie bazowej, te klasy z kolei w konstruktorze przyjmują trzy argumenty (dwa z nich są identyczne jak w klasie bazowej).

W klasach pochodnych zastosuj \*args, i \*\*kwargs.

Optymalizuj swój kod – wykorzystaj metodę super()

Podobnie zaimplementuj metodę w klasie bazowej i wywołaj ją dla obiektów będących instancjami obu klas.

W kolejnym kroku rozszerz liczbę argumentów w klasie bazowej oraz wybranej klasie pochodnej(w której zastosowałeś \*args i \*\*kwargs). Ile zmian należało wykonać aby ten krok przeprowadzić?