

Zadanie 1. Klasa Pizza (pol. Pizza)

1. Napisz klasę Pizza. Klasa powinna posiadać prywatne atrybuty instancyjne:
 1. price typu float (pol. cena)
 2. toppings typu list[str] (pol. dodatki)
 3. diameter typu float (pol. średnica pizzy)
2. Zaimplementuj:
 1. Inicjalizator z dwoma argumentami diameter oraz toppings. Cena zależy od powierzchni pizzy. Napisz statyczną metodę pomocniczą area liczącą powierzchnię pizzy. Powierzchnia pizzy powinna być obliczona wg wzoru: $\pi * (\text{diameter}/2)^2$. Cena powinna zostać ustawiona wg wzoru: $0.005 * \text{area} + \text{ilosc_dodatkow} * 2$. Średnica nie powinna być mniejsza niż 20. Jeśli będzie mniejsza niż 20 program powinien kończyć działanie z informacją „błędny promień” i kodem błędu -10.
 2. Zaimplementuj właściwość (propercję) setter dla diameter. Średnica nie powinna być mniejsza niż 20. Jeśli będzie mniejsza niż 20 program powinien kończyć działanie z informacją „błędna średnica” i kodem błędu -10. Jako, że od średnicy pizzy zależy jej cena, uaktualnij również wartość atrybutu cena.
3. Zaimplementuj metodę:
 1. add_topping przyjmującą jako argument topping (typu str). Metoda ta powinna dodawać do listy toppings podany dodatek oraz modyfikować cenę pizzy dodając do niej 2 za dodatek. Po każdym dodaniu dodatku, cena powinna być uaktualniana.
4. Nadpisz:
 1. Odpowiednią metodę magiczną, która zwróci reprezentację tekstową bieżącej instancji klasy:

„Pizza:

średnica: 30

dodatki: ser, szynka, pieczarki

cena: 60”

lub w razie braku dodatków:

„Pizza:

średnica: 30

cena: 30”

2. Metodę magiczną __add__, zwracającą nowy obiekt stworzony przez: wybranie z większych średnic, połączeniu list składników.

Zadanie 2. Klasa Slice (pol. Kawalek)

1. Zaimplementuj klasę Slice dziedziczącą po klasie Pizza. Klasa ta powinna posiadać atrybut instancyjny prywatny:
 1. `how_many_slices` (pol. Ile kawałków), typu `int`
2. Nadpisz inicjalizator nazywając dodatkowy argument `how_many_slices`. Ilość kawałków, na ile może być podzielona pizza nie może być mniejsza od 4 i większa od 12 i musi być parzysta. Jeśli `how_many_slices` będzie nieodpowiednie program powinien kończyć działanie z informacją „błędna ilość kawałków” i kodem błędu -10.
3. Zaimplementuj właściwości (propercje) setter i getter dla nowego atrybutu. Jeśli `how_many_slices` będzie nieodpowiednie program powinien kończyć działanie z informacją „błędna ilość kawałków” i kodem błędu -10.
4. Napisz metodę `part_price(ordered_slices)` która obliczy cenę `ordered_slices` zamówionych kawałków pizzy. Tj. jeżeli pizza może być podzielona na 6 kawałków i zamówimy 3 to cena powinna wynosić połowę całej pizzy.
5. Nadpisz metodę magiczną, która zwróci reprezentację tekstową bieżącej instancji klasy:

„Pizza:

średnica: 30

dodatki: ser, szynka, pieczarki

cena: 60

kawałki 6

cena za kawałek 10”

lub w razie braku dodatków:

„Pizza:

średnica: 30

cena: 30

kawałki 6

cena za kawałek 5”

Polecenia pomocnicze do testowania klas. Plik `main.py` z funkcją `main()` (Polecenia mają na celu pomoc w testowaniu napisanych klas)

Część do klasy Pizza:

1. Stwórz obiekt o nazwie `p_1`. Stwórz obiekt o nazwie `p_2`.
2. Wypisz na ekran reprezentację obiektów `p_1` i `p_2`.

3. Zmień wartość atrybutu diameter na 35 i dodaj jakiś dodatek i wypisz na ekran reprezentację obiektu p_1 po zmianie stanu.
4. Wypisz cenę pizzy p_1.
5. Dodaj obiekty p_1 i obiekt p_2. Wypisz wynik.

Część do klasy Slice:

1. Stwórz obiekt o nazwie s_1.
2. Wypisz na ekran reprezentację obiektu s_1.
3. Wypisz na ekran cenę za 5 kawałków s_1.