

Ćwiczenia 2

Programowanie obiektowe cz.2.

1. Utworzyć klasę `Tree` (drzewo), która będzie zawierała inicjalizator ustawiający następujące atrybuty:

- `name` (imię drzewa)
- `height` (wysokość drzewa [m])
- `leafs` (liczba liści),

oraz metody:

- `grow_up` (rośnij wzwyż), która przyjmie argument liczbowy w postaci wysokości do dodania, a następnie zwiększy wysokość tego drzewa
- `grow_wide` (rośnij wszcz), która przyjmie argument liczbowy w postaci liczby nowych liści,
- `show`, która wyświetli na ekranie wszystkie parametry drzewa wraz z ich wartościami

Utwórz 5 takich drzew (5 obiektów) i wyświetlić ich stan wewnętrzny, a następnie dla dwóch wybranych drzew zwiększyć wysokość i jeszcze raz wyświetlić ich stan wewnętrzny.

- Dodaj metodę magiczną `__str__` w powyższej klasie.

2. Stwórz klasę `Calendar` pełniącą funkcję kalendarza:

- dodaj w klasie konstruktor przyjmujący dwa argumenty odpowiadające za miesiąc i rok
- dodaj w klasie metodę `show` bez argumentu, która wyświetla w kolejnych dniach “kalendarz” od poniedziałku do niedzieli, max 7 dni, przykładowo dla października 2022

MO	TU	WE	TH	FR	SA	SU
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

3. Stwórz w programie dwie klasy `Pies` i `WlascicielPsa`. `Pies` musi być wyprowadzony na spacer jeśli spożywał posiłek więcej niż 5 godzin wcześniej. W klasie `Pies` dodaj metody:

- `UstawCzasPosilku(int)`, która ustawia ile godzin temu pies zjadł posiłek;
- `PobierzCzasPosilku()`, która pobiera czas posiłku;
- `PotrzebnySpacer()`, która zwraca `true` gdy pies musi być wyprowadzony na spacer.

W klasie `WlascicielPsa` dodaj metodę `ZabierzNaSpacer(Pies)`, która zwraca `true` gdy pies (będący parametrem metody) musi być wyprowadzony na spacer. Przetestuj działanie powyższych metod.

4. Stwórz klasę `Konto` z kilkoma polami, metodami i konstruktorami (saldo początkowe, końcowe, przelew między kontami, przelew zewnętrzny, wpłata, wypłata itd.) Następnie utwórz co najmniej 3 obiekty i wykonaj kilka metody na nich.

5. Utwórz klasę `Smartphone`. Konstruktor klasy powinien tworzyć i inicjalizować trzy prywatne atrybuty: `manufacturer`, `model` i `price`. Zdefiniuj w tej klasie metody ustawiające i wyświetlające wartości atrybutów.

Napisz program tworzący listę kilku obiektów klasy `Smartphone` i wyświetlający je.

6. Napisz klasę `Ułamek`, która przechowuje ułamki postaci $\frac{a}{b}$. Klasa przechowuje dwa pola: licznik i mianownik. Napisz metodę `skroc()`, która skraca ułamek, wymaga obliczenia największego wspólnego dzielnika.