

Wstęp do programowania obiektowego

Lista 7

Agnieszka Kazimierska, Karol Kulinowski

7 czerwca 2023

Lista nieobowiązkowa – poinformuj prowadzącego, jeżeli planujesz złożyć ją do sprawdzenia.

Każda klasa ma się znaleźć w oddzielnym pliku.

Podczas rozwiązywania zadań:

- Zwróć uwagę na typ zmiennych przechowywanych lub przyjmowanych przez klasy (łańcuchy znaków, liczby całkowite/zmiennoprzecinkowe, wartości logiczne, inne klasy).
- Zwróć uwagę na dopuszczalne wartości zmiennych przyjmowanych przez klasy. Jeżeli treść zadania nie określa *explicite* dopuszczalnych wartości, zastanów się, czy warunki nie wynikają z treści zadania. Zaimplementuj rzucanie i przechwytywanie wyjątków.
- Zwróć uwagę na opis pól klasy w poleceniu oraz podział funkcjonalności pomiędzy metody.

Zadanie 1

Napisz klasy reprezentujące różne rodzaje leków, wykorzystując mechanizm interfejsów.

- Interfejs *Lek* ma posiadać trzy metody:
 - metoda *obliczZapotrzebowanie()* ma przyjmować jeden argument reprezentujący czas trwania leczenia w dniach (liczba całkowita), a zwracać liczbę całkowitą,

- metoda *obliczCenę()* ma przyjmować jeden argument reprezentujący liczbę jednostek leku (liczba całkowita), a zwracać liczbę zmiennoprzecinkową,
 - metoda *podajDawkowanie()* ma nie przyjmować żadnych argumentów, a zwracać łańcuch znaków.
- Klasa *LekTabletki* ma posiadać pola reprezentujące dzienną dawkę leku w tabletkach (liczba całkowita), liczbę tabletek w opakowaniu (liczba całkowita) oraz cenę za opakowanie (liczba zmiennoprzecinkowa). Klasa ma implementować interfejs *Lek*, przy czym:
 - metoda *obliczZapotrzebowanie()* ma obliczać liczbę opakowań potrzebnych na cały czas leczenia na podstawie dziennej dawki, czasu trwania leczenia (podanego jako argument metody) oraz liczby tabletek w opakowaniu (zaokrąglij w górę do pełnych opakowań),
 - metoda *obliczCenę()* ma obliczać cenę określonej ilości leku jako iloczyn liczby opakowań (podanej jako argument metody) i ceny za opakowanie,
 - metoda *podajDawkowanie()* ma zwracać instrukcję dawkowania leku (na podstawie dziennej dawki) w formie łańcucha znaków, np. "Przyjmuj 3 razy na dzień" dla dziennej dawki równej 3.
 - Klasa *LekZastrzyk* ma posiadać pola reprezentujące dzienną liczbę zastrzyków (liczba całkowita) oraz cenę jednostkową zastrzyku (liczba zmiennoprzecinkowa). Klasa ma implementować interfejs *Lek*, przy czym:
 - metoda *obliczZapotrzebowanie()* ma obliczać liczbę zastrzyków potrzebnych na cały czas leczenia na podstawie dziennej liczby zastrzyków oraz czasu trwania leczenia (podanego jako argument metody),
 - metoda *obliczCenę()* ma obliczać cenę określonej liczby zastrzyków jako iloczyn liczby zastrzyków (podanej jako argument metody) i ceny za pojedynczy zastrzyk,
 - metoda *podajDawkowanie()* ma zwracać instrukcję przyjmowania zastrzyków (na podstawie dziennej liczby zastrzyków) w formie łańcucha znaków, np. "Wykonaj 3 razy na dzień" dla dziennej liczby zastrzyków równej 3.

- Klasa *AsystentLeczenia* ma posiadać pole typu *Lek* (tzn. taki sam typ jak interfejs) reprezentujące przyjmowany lek oraz trzy metody: *obliczZapotrzebowanie()*, *obliczCenę()* oraz *podajDawkowanie()*. Każda z tych metod ma wywoływać analogiczną metodę obiektu przechowywanego w polu reprezentującym lek (a więc też przyjmować i zwracać analogiczne argumenty).

Napisz program pokazujący działanie klasy *AsystentLeczenia* na przykładzie dla leku w tabletkach oraz leku w zastrzykach. Program nie musi przyjmować danych od użytkownika - możesz stworzyć obiekty z przykładowymi danymi “na sztywno” w kodzie.