Wstęp do programowania obiektowego
Lista 4

Agnieszka Kazimierska, Karol Kulinowski

19 kwietnia 2023

Termin składania rozwiązań:
14.05 (grupy wtorkowe)
09.05 (grupy środowe)
21.05 (grupa poniedziałkowa)

Programy przeznaczone do oceny należy umieszczać w repozytorium kursu w portalu eportal.pwr.edu.pl. Programy należy oddawać w formie plików źródłowych, bez pakowania.

Każda klasa ma się znaleźć w oddzielnym pliku. Rozwiązanie danego zadania ma się składać z pliku z klasą oraz oddzielnego pliku z programem demonstrującym jej działanie.

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.
- Pamiętaj o komentarzach dokumentacyjnych zgodnych z formatem Javadoc.

Zadanie 1

Napisz klasę publiczną reprezentującą wektor w dwuwymiarowym układzie współrzędnych kartezjańskich.

- Klasa ma przechowywać w polach współrzędne x, y końca wektora reprezentowane przez liczby zmiennoprzecinkowe. Początek wektora ma się znajdować w punkcie *(0,0)* układu współrzędnych.
- Klasa ma posiadać konstruktor publiczny przyjmujący współrzędne końca wektora.
- Pola klasy mają być zabezpieczone przed dostępem z zewnątrz. Dostęp do pól klasy ma się odbywać przez metody zwracające i ustawiające.
- Klasa ma posiadać metodę publiczną obliczającą długość wektora. Metoda nie ma przyjmować żadnych parametrów (tzn. ma korzystać tylko z wartości przechowywanych w polach klasy).
- Klasa ma posiadać metodę publiczną skalującą wektor (tzn. mnożącą obie współrzędne przez tę samą liczbę). Metoda ma przyjmować współczynnik skalowania jako liczbę całkowitą, a zwracać nowy obiekt klasy Wektor.
- Klasa ma posiadać metodę publiczną zwracającą reprezentację tekstową wektora w postaci `v = [vx, vy]`, gdzie `vx`, `vy` to współrzędne x, y końca wektora.

Napisz program pokazujący działanie klasy, w tym tworzenie obiektów klasy i działanie wszystkich jej metod. Program nie musi przyjmować danych od użytkownika - możesz stworzyć obiekty z przykładowymi danymi "na sztywno" w kodzie.

Zadanie 2

Napisz klasę publiczną reprezentującą urządzenie medyczne.

- Klasa ma przechowywać w polach numer seryjny urządzenia, rok produkcji, oddział, na którym znajduje się urządzenie oraz datę następnego przeglądu.[1]

[1]Sugestia: Wykorzystaj klasę *LocalDate* i jej metody *of()* (przy tworzeniu obiektów), *isAfter()*, *now()* (przy sprawdzaniu daty przeglądu).

- Klasa ma posiadać dwa konstruktory publiczne. Pierwszy konstruktor ma przyjmować wartości dla wszystkich pól klasy. Drugi konstruktor ma przyjmować wszystkie wartości poza datą następnego przeglądu i ustawiać ją na pustą (*null*). Zaimplementuj zabezpieczenia na sytuacje, gdy dane przekazane do konstruktora są nieprawidłowe (pusty numer seryjny, rok produkcji późniejszy niż obecny, pusta nazwa oddziału).

- Pola klasy mają być zabezpieczone przed dostępem z zewnątrz. Dostęp do pól klasy ma się odbywać przez metody zwracające i ustawiające.

- Klasa ma posiadać metodę publiczną do sprawdzania, czy przegląd jest ważny (tzn. czy data następnego przeglądu jest późniejsza niż data obecna). Metoda ma zwracać wartość logiczną. Uwzględnij sytuację, gdy urządzenie nie ma podanej daty następnego przeglądu.

- Klasa ma posiadać metodę publiczną zwracającą reprezentację tekstową urządzenia w postaci:
  `SN: `*`nr_seryjny`*` [prod. `*`rok`*`], `*`oddział`*`, przegląd `*`ważny/nieważny`*
  gdzie w miejsce wyrazów napisanych kursywą mają być wstawione informacje z pól danego obiektu (numer seryjny, rok produkcji, nazwa oddziału) lub wyznaczone za pomocą jego metod (przegląd ważny/nieważny w zależności od wyniku sprawdzania metodą z punktu poprzedniego).

Napisz program pokazujący działanie klasy, w tym tworzenie obiektów klasy i działanie wszystkich jej metod. Program nie musi przyjmować danych od użytkownika - możesz stworzyć obiekty z przykładowymi danymi "na sztywno" w kodzie.

Zadanie 3

Napisz klasę publiczną reprezentującą zbiór wszystkich urządzeń medycznych w szpitalu, wykorzystującą klasę z zad. 2.

- Klasa ma przechowywać w polach nazwę szpitala oraz listę urządzeń (tzn. listę obiektów klasy zdefiniowanej w zad. 2).

- Klasa ma posiadać konstruktor publiczny, który przyjmie nazwę szpitala oraz stworzy pustą listę urządzeń.

- Pola klasy mają być zabezpieczone przed dostępem z zewnątrz. Dostęp do pól klasy ma się odbywać przez metody zwracające i ustawiające.

- Klasa ma posiadać metodę publiczną dodającą do listy nowe urządzenie. Metoda ma przyjmować obiekt klasy zdefiniowanej w zad. 2.
- Klasa ma posiadać metodę publiczną usuwającą z listy urządzenie o podanym numerze seryjnym. Numer seryjny urządzenia ma być podawany jako argument metody. Uwzględnij sytuację, gdy na liście nie ma urządzenia o podanym numerze seryjnym.
- Klasa ma posiadać metodę publiczną zwracającą informację o wszystkich urządzeniach (tzn. wypisującą w konsoli reprezentację tekstową wszystkich urządzeń na liście). Uwzględnij sytuację, gdy lista urządzeń jest pusta.
- Klasa ma posiadać metodę publiczną zwracającą informację o urządzeniach, które mają nieważny przegląd (tzn. wypisującą w konsoli reprezentację tekstową tych urządzeń, których data przeglądu jest wcześniejsza niż obecna; do sprawdzania ważności przeglądu wykorzystaj metodę napisaną w zad. 2).

Napisz program pokazujący działanie klasy, w tym tworzenie obiektów klasy i działanie wszystkich jej metod. Program nie musi przyjmować danych od użytkownika - możesz stworzyć obiekty z przykładowymi danymi "na sztywno" w kodzie.

Zadanie 4 *(dodatkowe)*

Zmodyfikuj rozwiązania zad. 2 i 3 tak, aby obie klasy znajdowały się w tej samej paczce. Ustaw poziom dostępu do klas tak, aby były dostępne tylko w ramach paczki.