Wstęp do programowania obiektowego
Lista 5

Agnieszka Kazimierska, Karol Kulinowski

10 maja 2023

Termin składania rozwiązań:
30.05 (grupy wtorkowe)
24.05 (grupy środowe)
05.06 (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 plików z klasami oraz oddzielnego pliku z programem demonstrującym ich 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 klasy reprezentujące mierniki elektryczne stosowane w laboratorium.

- Klasa bazowa *Miernik* ma posiadać pola przechowujące informacje o modelu miernika (łańcuch znaków) i jego typie (łańcuch znaków; np. woltomierz, amperomierz itd.) oraz metodę zwracającą reprezentację tekstową miernika w postaci: `Typ_miernika [model]`
- Klasa *MiernikAnalogowy* ma dziedziczyć pola i metody klasy bazowej *Miernik*, a także:
  - posiadać dodatkowe pole przechowujące informację o klasie miernika,
  - posiadać dodatkową metodę *obliczNiepewnosc()*, przyjmującą jako parametr zakres miernika (liczba zmiennoprzecinkowa) i obliczającą niepewność bezwzględną pomiaru (liczba zmiennoprzecinkowa) ze wzoru:
$$\Delta_P = (klasa * zakres)/100 \tag{1}$$
- Klasa *MiernikCyfrowy* ma dziedziczyć pola i metody klasy bazowej *Miernik*, a także:
  - posiadać dodatkowe pola przechowujące informację o składowej multiplikatywnej δ_p i addytywnej Δ_z błędu pomiarowego miernika (na potrzeby zadania przyjmij, że miernik obsługuje tylko jeden zakres i składniki błędu pomiarowego są stałe),
  - posiadać dodatkową metodę *obliczNiepewnosc()*, przyjmującą jako parametr wynik pomiaru (liczba zmiennoprzecinkowa) i obliczającą niepewność bezwzględną pomiaru (liczba zmiennoprzecinkowa) ze wzoru:
$$\Delta_P = \delta_p * wynik\_pomiaru + \Delta_z \tag{2}$$

Napisz program pokazujący działanie wszystkich klas, 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 klasy reprezentujące różne rodzaje dokumentów cytowanych jako źródła przy pisaniu prac pisemnych.

- Klasa bazowa *Dokument* ma posiadać dwa pola, z których jedno ma przechowywać informację o autorze dokumentu (łańcuch znaków), a drugie o jego tytule (łańcuch znaków).

  Klasa ma posiadać metodę *wygenerujOdnosnik()* zwracającą reprezentację tekstową dokumentu w postaci: *Autor*: *Tytuł*. gdzie w miejsce wyrazów napisanych kursywą mają być wstawione informacje z pól obiektu.

- Klasa *Książka* ma dziedziczyć pola klasy bazowej *Dokument*, a także:

  - posiadać dodatkowe pola przechowujące informacje o roku wydania (liczba całkowita) oraz nazwie wydawcy (łańcuch znaków),
  - przesłaniać metodę *wygenerujOdnosnik()* w taki sposób, że dla książki reprezentacja tekstowa będzie miała postać:
    Autor: *Tytuł*. *Wydawca*, *rok_wydania*.
    gdzie w miejsce wyrazów napisanych kursywą mają być wstawione informacje z pól obiektu.

- Klasa *Rozdział* ma dziedziczyć pola klasy bazowej *Książka*, a także:

  - posiadać dodatkowe pola przechowujące informacje o redaktorze książki (łańcuch znaków) oraz tytule książki (łańcuch znaków),
  - przesłaniać metodę *wygenerujOdnosnik()* w taki sposób, że dla rozdziału książki reprezentacja tekstowa będzie miała postać:
    Autor: *Tytuł*. W: *Tytuł_książki*, red. *Redaktor*.
    Wydawca, *rok_wydania*.
    gdzie w miejsce wyrazów napisanych kursywą mają być wstawione informacje z pól obiektu.

Przy generowaniu reprezentacji tekstowych zwróć uwagę na dokładne formatowanie łańcucha znaków, w szczególności rodzaje i ułożenie znaków interpunkcyjnych. Napisz program pokazujący działanie wszystkich klas, 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.