

# Wstęp do programowania obiektowego

## Lista 3

Agnieszka Kazimierska, Karol Kulinowski

4 kwietnia 2023

Termin składania rozwiązań:

23.04 (grupy wtorkowe)

18.04 (grupy środowe)

30.04 (grupa poniedziałkowa)

Programy przeznaczone do oceny należy umieszczać w repozytorium kursu w portalu [eportal.pwr.edu.pl](http://eportal.pwr.edu.pl). Programy należy oddawać w formie plików źródłowych, bez pakowania. Podczas rozwiązywania zadań:

- Zwróć uwagę na typ zmiennych podawanych przez użytkownika ze standardowego wejścia (łańcuchy znaków, liczby całkowite/zmiennoprzecinkowe).
- Zwróć uwagę na dopuszczalne wartości zmiennych podawanych przez użytkownika. Jeżeli treść zadania nie określa *explicite* dopuszczalnych wartości, zastanów się, czy warunki nie wynikają z treści zadania. Zaimplementuj rzucanie i przechwytywanie wyjątków zgodnie z poleceniami.
- Zwróć uwagę na podział funkcjonalności pomiędzy metodę `main()` i dodatkowe metody własne, w szczególności: która metoda odpowiada za pobieranie danych od użytkownika, która metoda wykonuje obliczenia, a która ma wyświetlać wynik na standardowym wyjściu. Zwróć uwagę na to, jakie argumenty mają być przekazywane i jakie wartości mają być zwracane z metod własnych.
- Pamiętaj o komentarzach dokumentacyjnych dla metod własnych zgodnych z formatem Javadoc (w szczególności o krótkim opisie metody, podaniu parametrów wejściowych oraz wartości zwracanej).

## Zadanie 1

Napisz program do wyznaczania równania prostej na podstawie współrzędnych dwóch punktów leżących na tej prostej. Program ma spełniać następujące wymagania:

- Metoda `main()` ma pobierać od użytkownika współrzędne  $(x, y)$  dwóch punktów (załóż, że dane wejściowe są podawane jako liczby zmiennoprzecinkowe), a następnie wywoływać metodę do wyznaczania równania prostej.
- Metoda do wyznaczania równania prostej ma przyjmować współrzędne dwóch punktów jako argumenty wejściowe, obliczać współczynniki równania prostej, a następnie zwracać je w formie dwuelementowej tablicy liczb zmiennoprzecinkowych. Jeżeli podane wartości nie pozwalają na obliczenie współczynników prostej (np. spowodują błąd), metoda ma rzucać wyjątek wybranego typu z odpowiednim komunikatem.
- Po wykonaniu obliczeń metoda `main()` ma wyświetlać wynik na standardowym wyjściu w formie łańcucha znaków, np.  $y = a * x + b$ , gdzie w miejsce  $a$ ,  $b$  są wstawione obliczone współczynniki równania.

## Zadanie 2

Napisz program do wyświetlania sformatowanej daty na podstawie numeru dnia, miesiąca i roku. Program ma spełniać następujące wymagania:

- Metoda `main()` ma pobierać od użytkownika dzień, miesiąc oraz rok (załóż, że dane wejściowe są podawane jako liczby całkowite), a następnie wywoływać metodę do formatowania daty.
- Metoda do formatowania daty ma przyjmować dzień, miesiąc i rok jako argumenty wejściowe, a następnie sprawdzać poprawność podanych danych, np. czy numer miesiąca pochodzi z dopuszczalnego zakresu oraz czy numer dnia pochodzi z zakresu dopuszczalnego dla danego miesiąca (możesz pominąć zmiany związane z latami przestępnymi).
- Jeżeli któraś z wartości jest nieprawidłowa, metoda ma rzucać wyjątek typu `IllegalArgumentException` z odpowiednim komunikatem.

- Jeżeli wartości przekazane do metody są prawidłowe, metoda ma formatować datę do zapisu: *DD-MM-RRRR*, gdzie *DD* oznacza dzień, *MM* – miesiąc, *RRRR* – rok, a następnie zwracać sformatowany łańcuch znaków. Numer dnia i miesiąca powinien być zapisany na dwóch cyfrach (zaimplementuj dodawanie 0 na początku tam, gdzie jest potrzebne).
- Metoda `main()` ma obsługiwać za pomocą bloku `try-catch` wyjątki zwracane z metody do formatowania daty. Jeżeli zostanie przechwycony wyjątek, metoda `main()` ma wyświetlić odpowiedni komunikat na standardowym wyjściu. Jeżeli nie zostanie przechwycony wyjątek, metoda `main()` ma wyświetlać sformatowaną datę na standardowym wyjściu.

## Zadanie 3

Napisz program do obliczania pola powierzchni trzech figur: koła, prostokąta i trapezu. Program ma spełniać następujące wymagania:

- Metoda `main()` ma przyjmować od użytkownika informację o typie figury (oznaczoną np. liczbą 1, 2, 3 lub różnymi łańcuchami znaków) oraz odpowiednią liczbę parametrów potrzebnych do obliczenia pola danej figury. Załóż, że dane wejściowe są podawane jako liczby zmiennoprzecinkowe.
- Metody obliczające pola figur mają być napisane z wykorzystaniem mechanizmu przeładowania metod, tzn. mieć tę samą nazwę, ale różnić się liczbą przyjmowanych parametrów. Metody obliczające pola mają zwracać obliczoną wartość pola.
- Metody obliczające pola mają sprawdzać poprawność podanych danych i rzucać wyjątki, jeżeli któraś z wartości jest nieprawidłowa (analogicznie jak w zad. 1 i 2).
- Metoda `main()` ma wyświetlać obliczone pole na standardowym wyjściu z dokładnością do dwóch miejsc po przecinku. Metoda `main()` ma obsługiwać za pomocą bloków `try-catch` wyjątki zwracane z metod do obliczania pól i wyświetlać odpowiedni komunikat, jeżeli zostanie przechwycony wyjątek (analogicznie jak w zad. 1 i 2).

## Zadanie 4

Napisz program do tworzenia listy nazwisk studentów wykorzystujący tablicę `ArrayList`. Program ma spełniać następujące wymagania:

- Użytkownik ma mieć możliwość podawania nazwisk w pętli aż do podania wartości kończącej działanie programu (np. liczby 0 czy napisu *stop*).
- Program ma sprawdzać, czy podane nazwisko zawiera tylko litery lub znak myślnika. Jeżeli tak, podane nazwisko ma być dodane do listy. Jeżeli nie, program ma wyświetlić komunikat o nieprawidłowym formacie danych, ale nie przerywać pętli.
- Po zakończeniu dodawania program ma posortować listę nazwisk w kolejności alfabetycznej i wyświetlić na standardowym wyjściu informację o całkowitej liczbie studentów oraz posortowaną listę nazwisk.
- Każde nazwisko ma być wyświetlone wraz z numerem na posortowanej liście, np.
  1. Adamski
  2. Branicki
  3. Czyż
  4. Dudek