

JAVA - PODSTAWY: PROGRAMOWANIE

ZADANIE 1.

Napisz program, który pobierze od użytkownika średnicę okręgu (zmienną typu float) i obliczy obwód tego okręgu. Do obliczeń przyjmij najpierw $\pi = 3.14$, a następnie skorzystaj z wbudowanej klasy Math i znajdującej się tam stałej PI.

ZADANIE 2.

Napisz program obliczający BMI (Body Mass Index) i sprawdzający, czy jest ono w normie. Program powinien wczytywać od użytkownika dwie zmienne: wagę w kilogramach (typ float) oraz wzrost w centymetrach (typ int). BMI powinno zostać wyliczone zgodnie z poniższym wzorem:

$$BMI = \frac{\text{masa [w kilogramach]}}{\text{wzrost[w metrach]}^2}$$

Optymalny zakres BMI wynosi od 18.5 do 24.9, wartości mniejsze lub większe są wartościami nieoptymalnymi. Twój program powinien wypisać „BMI optymalne” lub „BMI nieoptymalne”, zgodnie z powyższymi założeniami.

ZADANIE 3.

Napisz program służący do rozwiązywania równania kwadratowego. Program powinien pobrać trzy liczby całkowite (współczynniki równania kwadratowego a , b , c) i wyliczyć pierwiastki x_1 , x_2 równania

$$ax^2 + bx + c = 0$$

.

Jeśli delta Δ wyjdzie ujemna, wypisz „Delta ujemna” i zakończ program.

Potrzebne wzory:

$$\begin{aligned}\Delta &= b^2 - 4ac \\ x_1 &= \frac{-b - \sqrt{\Delta}}{2a} \\ x_2 &= \frac{-b + \sqrt{\Delta}}{2a}\end{aligned}$$

ZADANIE 4.

Napisz program, który pobierze od użytkownika dodatnią liczbę (typu int) i wypisze wszystkie liczby od 1 do podanej liczby, każda w kolejnej linii, z następującymi zmianami:

- w miejscu liczb podzielnych przez 3, zamiast liczby program powinien wypisać „Pif”
- w miejscu liczb podzielnych przez 7, zamiast liczby program powinien wypisać „Paf”
- jeśli liczba jest podzielna zarówno przez 3, jak i 7, program powinien wypisać „Pif paf”

ZADANIE 5.

Napisz program, który pobierze od użytkownika liczbę dodatnią (typu *int*) i wypisze wszystkie liczby pierwsze, większe od 1 i mniejsze od podanej liczby.

ZADANIE 6.

Napisz program, który pobierze od użytkownika liczbę n (typu *int*) i obliczy sumę szeregu harmonicznego od 1 do n , zgodnie ze wzorem poniżej:

$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$$

ZADANIE 7.

Napisz program, który pobierze od użytkownika liczbę dodatnią (typu *int*) i obliczy liczbę Fibonacciego o wskazanym indeksie. Przykładowo, jeśli użytkownik poda liczbę 5, Twój program powinien wypisać piątą liczbę Fibonacciego. Kolejne liczby Fibonacciego powstają poprzez zsumowanie dwóch poprzednich liczb Fibonacciego. Przykładowo, kilka pierwszych liczb Fibonacciego wynosi kolejno:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377...

ZADANIE 8.

Napisz program realizujący prosty kalkulator.

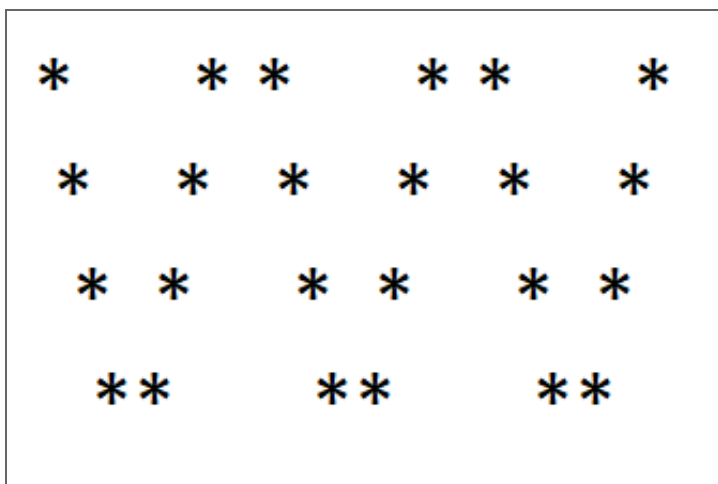
Program powinien:

- a. pobrać pierwszą liczbę (typu float)
- b. pobrać jeden ze znaków: + - / *
- c. pobrać drugą liczbę (typu float)
- d. zwrócić wynik pobranego działania

Jeśli użytkownik poda znak inny niż obsługiwane, program powinien wypisać „Błędny znak”. Jeśli wpisanego działania nie da się zrealizować (tj. jest niezgodne z zasadami matematyki), to program powinien wypisać napis „Błąd”.

ZADANIE 9.

Napisz program, który pobierze od użytkownika liczbę dodatnią (typu int) i narysuje falę o zadanej długości i wysokości 4 linijek, zgodnie z poniższym schematem (puste pola uzupełniaj spacjami):



ZADANIE 10.

Napisz program, który pobierze od użytkownika jedną liczbę dodatnią (typu int) i obliczy sumę cyfr podanej liczby. Podpowiedź: aby rozpatrywać liczbę cyfra po cyfrze, możesz obliczać resztę z dzielenia liczby przez 10 (aby uzyskać wartość ostatniej cyfry) i dzielić liczbę bez reszty przez 10 (żeby „przesuwać” się do kolejnej cyfry).

ZADANIE 11.

Napisz program, który będzie od użytkownika pobierał kolejne teksty (zmienne typu String) tak długo, aż użytkownik poda tekst „Starczy”, a następnie wypisze najdłuższy z podanych tekstów (nie biorąc pod uwagę tekstu „Starczy”). Jeśli użytkownik nie poda żadnego tekstu, to wypisz „Nie podano żadnego tekstu”.

ZADANIE 12.

Napisz program, który pobierze od użytkownika tekst (zmienną typu String) i policzy jakim procentem wszystkich znaków tekstu był znak spacji, czyli

$$\frac{\textit{liczba spacji}}{\textit{liczba wszystkich znaków}} * 100\%$$

ZADANIE 13.

Napisz program, który „się jąka”, to znaczy pobiera użytkownika tekst (zmienną typu String), a następnie wypisuje podany tekst, w którym każde słowo wypisane jest po dwa razy. Przykładowo, dla wejścia: „To jest mój test” program powinien wypisać „*To To jest jest mój mój test test*”.

ZADANIE 14.

Napisz program, który pobierze od użytkownika dwie małe litery alfabetu łacińskiego (typ char) i wyliczy, ile znaków stoi w alfabecie pomiędzy podanymi literami. Podpowiedź – skorzystaj z tablicy kodów ASCII i traktuj znaki jak liczby int.

ZADANIE 15.

Napisz program, który pobierze od użytkownika 10 dowolnie dużych liczb (zmiennych typu *int*) i wypisze te, które wystąpiły minimum dwukrotnie.

ZADANIE 16.

Napisz program, który pobierze od użytkownika 10 liczb (zmiennych typu int) i wypisze długość najdłuższego takiego podciągu tych liczb, który jest rosnący. Przykładowo, dla liczb: „1, 3, 8, 4, 2, 5, 6, 11, 13, 7” program powinien wypisać „5” jako długość najdłuższego rosnącego podciągu (w przykładzie podkreślonego).

ZADANIE 17.

Napisz program, który pobierze od użytkownika datę najbliższych Twoich zajęć w SDA i obliczy ile dni do nich pozostało. Podpowiedź: datę wczytaj jako typ **String** i przeparsuj na **LocalDate**. Obecną datę pobierz z metody **LocalDate.now()**.

ZADANIE 18.

Napisz program, który pobierze od użytkownika tekst (zmienna typu **String**) i sprawdzi, czy użytkownik kichnął, tzn. czy w podanym tekście znajduje się tekst „**aaaa psik**” z dowolnie wieloma, ale minimum jedną literą ‘a’ na początku wyrażenia (czyli kichnięciem jest zarówno „a psik”, jak i „aaaaaaaaaaa psik”). Podpowiedź: użyj wyrażenia regularnego z odpowiednim kwantyfikatorem.

ZADANE 19.

Napisz program, składający się z kilku klas:

- a. Klasy *Author*, reprezentującą autora – pisarza wierszów, składającej się z pól *surname* oraz *nationality* (oba typu String)
- b. Klasy *Poem*, reprezentującą wiersz, składającą się z pól *creator* (typu Author) oraz *stropheNumbers* (typu int – reprezentującej ilość zwrotek wiersza)
- c. Klasy *Main*, w której znajdzie się metoda *main*, a w niej kod w którym:
 - i. Stworzysz trzy obiekty klasy *Poem*, uzupełniając go wszystkimi danymi (używając konstruktora lub *setter'ów*) i zapiszesz je w tablicy
 - ii. Wypiszesz nazwisko tego autora, który napisał wiersz o największej liczbie zwrotek (niech odpowiedź obliczy Twój program!)

ZADANIE 20.

Napisz program, który będzie z Tobą grał w „za dużo, za mało”. Na początku komputer losuje liczbę z zakresu od 0 do 100 (podpowiedź: skorzystaj z metody **Random.nextInt()**) a następnie czeka na podanie liczby przez użytkownika. Jeśli użytkownik poda liczbę większą, niż wylosowana przez komputer, program powinien wypisać „**za dużo**” i czekać na podanie kolejnej liczby. Jeśli użytkownik poda liczbę mniejszą, program powinien wypisać „**za mało**” i analogicznie czekać na następną liczbę. Jeśli użytkownik poda odpowiednią wartość, to program powinien wypisać słowo „**Bingo!**” i zakończyć się.