

## Wprowadzenie do Javy – zadania dodatkowe

Podstawy:

1. Zadeklaruj trzy zmienne liczbowe (całkowite) i przypisz im wartości. Następnie wypisz wartość największej z nich.
2. Sprawdź, czy wartość przypisana zmiennej jest parzysta, czy nie (wypisz na ekranie odpowiedni komunikat). Użyj google, aby znaleźć sposób na sprawdzenie parzystości wartości.
3. Sprawdź, czy wartość zmiennej jest większa, mniejsza, czy równa zero (użyj google żeby znaleźć odpowiedź, jak obsłużyć więcej niż dwa przypadki w instrukcji warunkowej).
4. Rozwiąż zadanie 1. dla czterech zmiennych.

1. Zadeklaruj trzy zmienne liczbowe (całkowite) i przypisz im wartości. Następnie wypisz wartość największej z nich.
2. Sprawdź, czy wartość przypisana zmiennej jest parzysta, czy nie (wypisz na ekranie odpowiedni komunikat). Użyj google, aby znaleźć sposób na sprawdzenie parzystości wartości.
3. Sprawdź, czy wartość zmiennej jest większa, mniejsza, czy równa zero (użyj google żeby znaleźć odpowiedź, jak obsłużyć więcej niż dwa przypadki w instrukcji warunkowej).
4. Rozwiąż zadanie 1. dla czterech zmiennych.

For:

1. Wypisz wszystkie liczby z zakresu 5 - 100.
2. Wypisz wszystkie liczby z zakresu 55 - 150.
3. Wypisz wszystkie liczby z zakresu 1 - 100 zaczynając od 100.
4. Wypisz wszystkie liczby parzyste z przedziału 0 - 100.
5. Dla liczb z przedziału 0 - 100 podaj informację, czy są podzielne przez 2, 3 i 5, w następujący sposób:  
0, podzielne przez 2, podzielne przez 3, podzielne przez 5  
1  
2, podzielne przez 2  
3, podzielne przez 3  
4, podzielne przez 2  
5, podzielne przez 5  
6, podzielne przez 2, podzielne przez 3  
7  
8, podzielne przez 2  
9, podzielne przez 3  
10, podzielne przez 2, podzielne przez 5  
...  
6. Podaj sumę wszystkich liczb z przedziału 1 - 459 (skorzystaj z pętli for).
7. Podaj wartość sumy wyrazów ciągu arytmetycznego. Wartość pierwszego wyrazu ciągu to 5, różnica każdych kolejnych dwóch wyrazów to 2. Chcemy zsumować 459 wyrazów (skorzystaj z pętli for).
8. Wypisz w konsoli:  
\*  
\*\*  
\*\*\*  
\*\*\*\*  
\*\*\*\*\*

Wierszy w takim formacie chcemy uzyskać tyle, ile wynosi wartość przypisana zmiennej x.



Odwiedź nas

Pomorski Park Naukowo Technologiczny, II piętro  
Al. Zwycięstwa 96/98  
81-451 Gdynia



Zadzwoń do nas

Telefon: 58 782 48 88  
Poniedziałek - Piątek, 8:00 - 16:00



Napisz do nas

Sprawy ogólne: info@sdacademy.pl  
Trenerzy: trenerzy@sdacademy.pl

www.sdacademy.pl



facebook.com/sdacademypl/



9. Wypisz w konsoli:

```
****
***
**
*
```

Analogicznie do 8. korzystając z wartości zmiennej x.

10. Podobnie jak poprzednie ale chcemy uzyskać trójkąt z kątem prostokątnym w górnym, prawym rogu:

```
****
***
**
*
```

char, charAt

1. Wypisz wszystkie litery angielskiego alfabetu.
2. Wypisz co drugi znak angielskiego alfabetu.
3. Wypisz znaki angielskiego alfabetu na wstak.
4. Znajdź największą literę w tekście "alfabet".
5. Zadeklaruj zmienną tekstową i przypisz jej wartość. Wypisz wszystkie znaki wartości tej zmiennej w osobnych wierszach.
6. Podaj liczbę wystąpień litery "a" w tekście "alfabet".
7. Sprawdź, czy tekst jest palindromem (czy czytany od tyłu będzie taki sam).
8. Sprawdź dla wybranego tekstu, dla której litery będziemy mieli najdłuższy ciąg wystąpień tylko tej litery. Przykładowo: aabbbccbb - dla tego tekstu będzie to b, które wystąpiło pod rząd 3 razy.
9. Zadeklaruj i przypisz wartości dwóm zmiennym tekstowym. Następnie sprawdź, czy pierwszy tekst zawiera drugi.

Metody

1. Napisz metodę multiply, która przyjmie dwa parametry typu int, a która zwróci wynik mnożenia tych parametrów.
2. Napisz metodę concatenate, która przyjmie dwa łańcuchy znaków jako parametry, a która zwróci ich konkatenację.
3. Napisz metodę multiply2, która zadziała tak jak w metoda multiply z zadania 1., ale wynik policzy z wykorzystaniem wielokrotnego dodawania.
4. Napisz metodę power, która przyjmie dwa parametry typu int, a która zwróci wynik potęgowania o podstawie i wykładniku takich, jak przekazano.
5. Napisz metodę, która przyjmie jako pierwszy parametr wartość typu String, a jako drugi znak. Metoda zwróci liczbę wystąpień znaku w łańcuchu.
6. Napisz metodę contains, która przyjmie dwa łańcuchy jako parametry, a która zwraca informację, czy pierwszy łańcuch zawiera drugi.

Tablice

1. Napisz metodę print, która wypisze w konsoli elementy tablicy intów, przekazanej jako parametr.
2. Napisz metodę getMax, która przyjmie za parametr tablicę intów, a która zwróci największą wartość z tej tablicy.
3. Napisz metodę getMin, która zwróci najmniejszy element przekazanej tablicy intów.
4. Napisz metodę getSum, która przyjmie za parametr tablicę intów, a która zwróci sumę jej elementów.
5. Napisz metodę getMaxMinAndSum, która przyjmie za parametr tablicę intów, a która zwróci tablicę trzelementową, której pierwszy element to wartość największa, drugi to najmniejsza, a trzeci suma wszystkich elementów.
6. Napisz metodę getLarger, która przyjmie za parametry dwie tablice intów, a która zwróci tę tablicę, której suma elementów jest większa.
7. Napisz metodę merge, która za parametry dwie tablice intów, a która zwróci tablicę, która zawierać będzie wszystkie elementy z jednej i drugiej tablicy.



Odwiedź nas

Pomorski Park Naukowo Technologiczny, II piętro  
Al. Zwycięstwa 96/98  
81-451 Gdynia



Zadzwoń do nas

Telefon: 58 782 48 88  
Poniedziałek - Piątek, 8:00 - 16:00



Napisz do nas

Sprawy ogólne: info@sdacademy.pl  
Trenerzy: trenerzy@sdacademy.pl

www.sdacademy.pl



facebook.com/sdacademypl/



8. Napisz metodę `getDistinct`, która za parametr przyjmie tablicę, a która zwróci tablicę, w której będą elementy bez powtórzeń z tablicy przekazanej w parametrze.

Scanner, switch

1. Stwórz program, który poprosi użytkownika o podanie wartości liczbowej. Następnie program wypisze wszystkie liczby naturalne, aż do zadanej.
2. Stwórz program, który poprosi użytkownika o podanie dwóch wartości - długości boków prostokąta. Następnie wypisze pole takiego prostokąta.
3. Stwórz program, który spyta użytkownika o podanie figury, której pole chce policzyć: kwadratu, prostokąta, koła czy trójkąta. Następnie zgodnie z wyborem zada odpowiednie pytania o wymiary figury i wypisze jej pole oraz obwód.
4. Stwórz program, który pyta użytkownika o wartość całkowitą i który daje odpowiedź na pytanie, czy podano liczbę pierwszą.

While

1. Stwórz program, który będzie zadawał użytkownikowi pytanie "Czy chcesz zakończyć działanie programu", póki nie odpowie "tak". (google: java porównanie stringów/java String comparison)
2. Stwórz program, który poprosi użytkownika o podanie wartości liczbowej. Następnie sprawdzi, która potęga liczby 3 jest większa od zadanej wartości. Np. jeśli użytkownik podał 75, to odpowiedzią będzie 4, bo 3 do czwartej potęgi jest równe 81 i to pierwsza potęga liczby 3 większa od 75.
3. Zapytaj użytkownika o liczbę całkowitą (kwotę). Następnie odpowiedz mu na pytanie, jaka kombinacja nominałów zajmie najmniej miejsca w portfelu. Np. dla 723 odpowiedzią jest:  
3 banknoty dwustuzłotowe, 1 banknot stużłotowy, jeden dwudziestizłotowy, moneta dwuzłotowa oraz złotówka.

Klasy

Do każdego zadania należy utworzyć kilka obiektów i przetestować działanie metod!

1. Stwórz klasę `Rectangle`, która będzie miała pola: `a`, `b` (długości boków). Do klasy dodaj konstruktor (ustawiający wartości wszystkich pól).
- 1b. Dodaj do klasy metodę `getArea`, która zwróci pole powierzchni, oraz metodę `getPerimeter` zwracającą obwód figury.
2. Postąp analogicznie dla koła oraz trójkąta (`a`, `b`, `c`).
3. Stwórz klasę `Person`, która będzie miała pola: `firstName`, `lastName` oraz `age` oraz konstruktor (ustawiający wartości wszystkich pól).
4. Stwórz klasę `Room`, która będzie miała pola: `area` oraz `height`.
5. Stwórz klasę `House`, która będzie miała pola: `address` (`String`), `kitchen`, `bathroom` oraz tablicę pokoi (`rooms`). W przypadku pól oznaczających pomieszczenia skorzystajmy z klasy `Room`.
6. Do klasy `House` dodaj metodę `getArea`, która zwróci pole powierzchni domu.
7. Stwórz klasę `Party`, która będzie miała pola: `host` (gospodarz), `attendees` (uczestnicy). Do klasy dodaj metodę `getAverageAge`, która zwróci średni wiek uczestników.
8. Stwórz typ `Lock`, reprezentujący zamek na kod (z trzema przekładniami - np. zamek do walizki/roweru). Będziemy mogli tworzyć obiekt typu `Lock` przekazując jako parametr konstruktora trzy cyfry - kod otwierający zamek. Domyślnie obiekt będzie tworzony z losowym ustawieniem przekładni. Obiekt typu `Lock` będzie oferował trzy metody zmieniający ustawienie każdej z przekładni o jedną pozycję w prawo. Będzie również oferował metodę sprawdzającą, czy zamek jest otwarty, czy nie, oraz metodę wypisującą na ekranie aktualnie ustawioną kombinację zamka.
9. Stwórz interfejs użytkownika dla zamka. Kolejno:



Odwiedź nas

Pomorski Park Naukowo Technologiczny, II piętro  
Al. Zwycięstwa 96/98  
81-451 Gdynia



Zadzwoń do nas

Telefon: 58 782 48 88  
Poniedziałek - Piątek, 8:00 - 16:00



Napisz do nas

Sprawy ogólne: [info@sdacademy.pl](mailto:info@sdacademy.pl)  
Trenerzy: [trenerzy@sdacademy.pl](mailto:trenerzy@sdacademy.pl)

[www.sdacademy.pl](http://www.sdacademy.pl)



[facebook.com/sdacademy/](https://facebook.com/sdacademy/)



- pytamy użytkownika w trzech krokach o poprawną kombinację
  - następnie użytkownik może:
    - sprawdzić aktualną kombinację zamka
    - przesunąć wybraną przekładnię
    - sprawdzić, czy zamek jest otwarty
10. Zmodyfikuj typ Lock, aby zamek mógł posiadać dowolną liczbę przekładni (zaktualizuj też oczywiście interfejs - użytkownik na początku określa ile jest przekładni).

#### Dziedziczenie:

1. Stwórz klasę Vehicle, która będzie miała pola speed (aktualna prędkość) oraz maxSpeed (prędkość maksymalna) typu int, konstruktor ustawiający wartość tych pól oraz getter i setter wspomnianych pól.
2. Stwórz klasę Car oraz klasę Bicycle, które będą rozszerzały klasę Vehicle. Do obydwu klas należy dodać pole, które będzie unikatowe dla danej klasy (np. rozmiar kół/liczba przerzutek, pojemność silnika/liczba drzwi/coś innego opisującego dany pojazd). Dodaj konstruktory ustawiające wartości nowych pól oraz ustawiające wartość pól speed oraz maxSpeed korzystając z konstruktora klasy Vehicle (przy użyciu słowa super).
3. W klasie Vehicle nadpisz metodę toString odziedziczoną z klasy Object w taki sposób, aby jej wynik opisywał wartości wszystkich pól.
4. Nadpisz metodę toString także w klasach Bicycle oraz Car w taki sposób, aby zwracały to, co zwracane jest dla pojazdów, z dodatkową informacją o wartościach pól zdefiniowanych w tych klasach.
5. Dodaj do klasy Vehicle pole typu int o nazwie wheelCount. Spraw, aby tworząc rower z marszu było ustawiane 2, a tworząc samochód było to pole inicjalizowane wartością 4. Uwzględnij informację o liczbie kół w metodzie toString().
6. Stwórz klasy reprezentujące kręgowce i bezkręgowce, następnie klasy reprezentujące ssaki, gady, płazy i ptaki. Dodaj po klasie, która będzie dziedziczyła po każdej z tych klas.

#### Inne:

7. Napisz metodę statyczną, przyjmującą za parametry łańcuch znaków i znak (liczbę) typu char. Funkcja powinna zwrócić łańcuch znaków, który będzie wynikiem zaszyfrowania przekazanego jako parametr tekstu przy użyciu szyfru Cezara (przesunięcie wyznacza drugi parametr).
8. Napisz metodę statyczną getPrimes, która przyjmie za parametr liczbę całkowitą, a która zwróci wszystkie mniejsze od niej liczby pierwsze, przy wykorzystaniu Sita Eratonesesa.
9. Napisz metodę, która zwróci rozkład liczby na czynniki pierwsze (z uwzględnieniem 1). Przykładowo dla 26 metoda zwróci: 1, 2, 13 (bo  $1 \cdot 2 \cdot 13 = 26$ )  
Dla 8 metoda zwróci: 1, 2, 2, 2. (bo  $1 \cdot 2 \cdot 2 \cdot 2 = 8$ ).
10. Napisz metodę statyczną, która przyjmie jako parametry dwie tablice intów. Metoda zwróci część wspólną - tablicę, która będzie zawierała elementy, które występują zarówno w pierwszej, jak i drugiej tablicy. Przykładowo dla: [1, 2, 2, 3, 4] i [0, 2, 2, 4, 5] metoda zwróci [2, 2, 4].
11. Napisz metodę, która zwróci najmniejszą wspólną wielokrotność z wykorzystaniem poprzednich dwóch metod.
12. Obdarowywanie:
  - 1) Obdarowująca osoba przez trzy kolejne dni obdarowuje po jednej osobie.
  - 2) Obdarowana osoba staje się osobą obdarowującą
  - 3) Zerowego dnia mamy jedną osobę obdarowującą.Po ilu dniach wszyscy na Ziemi zostaną obdarowani?

1. Stworzymy abstrakcyjną klasę Shape, która będzie miała pole:  
private String color;  
oraz metodę abstrakcyjną getArea(); (pole powierzchni)  
oraz metodę abstrakcyjną getPerimeter(); (obwód)



#### Odwiedź nas

Pomorski Park Naukowo Technologiczny, II piętro  
Al. Zwycięstwa 96/98  
81-451 Gdynia



#### Zadzwoń do nas

Telefon: 58 782 48 88  
Poniedziałek - Piątek, 8:00 - 16:00



#### Napisz do nas

Sprawy ogólne: info@sdacademy.pl  
Trenerzy: trenerzy@sdacademy.pl

www.sdacademy.pl



facebook.com/sdacademypl/





2. Tworzymy klasę Rectangle o dwóch polach typu int (reprezentujące długości boków), która będzie rozszerzała klasę Shape
3. Analogicznie tworzymy klasę Circle
4. Tworzymy klasę Square, która będzie rozszerzała klasę Rectangle.
5. Tworzymy interfejs Solid (Bryła) o metodach abstrakcyjnych getSurfaceArea oraz getVolume.
6. Tworzymy klasy w taki sposób, aby uwzględnić bryły obrotowe:  
Stożek oraz Walec
7. graniastoslupy:  
Prostopadłościan oraz Sześcian
8. Stwórz klasę PerpendicularSolid, która będzie reprezentowała bryłę o dwóch równoległych, jednakowych podstawach, której ściany będą prostopadłe do podstaw. Użyj jako podstaw obiektu typu Shape.
9. Stwórz abstrakcyjną klasę Animal, w której będzie zadeklarowana metoda:  
public void makeASound()  
Stwórz klasy Dog, Cat oraz Mouse rozszerzające tę klasę.
10. Stwórz interfejs Product (produkt w sklepie) z metodami:  
int getPrice();  
int getWeight();
11. Stwórz klasę ChocolateBar, która będzie reprezentowała batonik. Batonik ma pola cena i waga. Niech batonik będzie produktem (implementuje interfejs Product).
12. Stwórz klasę Milk, która reprezentuje mleko. Mleko ma pola: objętość, gęstość, oraz cena za 100 gram. Niech mleko także będzie produktem.

Generyczne:

1. Stwórz klasę generyczną Pair, która przechowuje dwie wartości, wybranych typów.

Generyczne:

2. Stwórz klasę Map, która będzie kontenerem słownikowym - kluczowi mamy przypisaną wartość. Klasa powinna posiadać metodę put, która przyjmuje dwa parametry: klucz i wartość. Powinna także posiadać metodę get, która przyjmuje jeden parametr - klucz, a zwraca wartość. Dodatkowo dopisz metodę remove, która jako parametr przyjmuje klucz, a która usuwa parę klucz - wartość dla danego klucza.
3. Stwórz klasę Set, która będzie reprezentowała zbiór. Zbiór od listy różni się tym, że element w zbiorze nie może wystąpić dwa razy. Dodatkowo kolejność elementów nie ma znaczenia. Powinna mieć metody add, size, contains oraz toArray.

Kolekcje:

Tworzymy metody statyczne:

1. Tworzymy metodę zadanie1, która przyjmie za parametr listę, której elementy są typu Integer, a która wypisze wszystkie elementy listy.
2. Tworzymy metodę zadanie2, która przyjmie za parametr listę, której elementy są typu Integer, a która wypisze wszystkie elementy listy wraz z ich indeksami.
3. Tworzymy metodę zadanie3, która przyjmie za parametr listę, której elementy są typu Integer, a która zwróci średnią wartości z tej listy.
4. Tworzymy metodę zadanie4, która przyjmie za parametr listę Stringów, a która zwróci odpowiedź na pytanie, czy lista zawiera łańcuch znaków, który rozpoczyna się od znaków "abc".
5. Tworzymy metodę zadanie5, która przyjmie za parametr listę Stringów, a która zwróci odpowiedź na pytanie, ile elementów ma nieparzystą liczbę znaków.
6. Tworzymy metodę zadanie6, która przyjmie za parametr zbiór Stringów, a która zwróci odpowiedź na pytanie, czy w zbiorze mamy łańcuch znaków, który kończy się na tekst "cde".
7. Tworzymy metodę zadanie7, która za parametr przyjmie listę Stringów, a zwróci zbiór, zawierający elementy tej listy.
8. Tworzymy metodę zadanie8, która za parametr przyjmie zbiór Stringów, a zwróci listę, zawierającą elementy tego zbioru.
9. Tworzymy klasę Converter, która będzie typem generycznym, która będzie miała w generyczny sposób napisane metody z zadań 7 oraz 8 (nazwijmy je toSet oraz toList).



Odwiedź nas

Pomorski Park Naukowo Technologiczny, II piętro  
Al. Zwycięstwa 96/98  
81-451 Gdynia



Zadzwoń do nas

Telefon: 58 782 48 88  
Poniedziałek - Piątek, 8:00 - 16:00



Napisz do nas

Sprawy ogólne: info@sdacademy.pl  
Trenerzy: trenerzy@sdacademy.pl

www.sdacademy.pl



facebook.com/sdacademypl/



10. W klasie Converter zamieniamy metody na statyczne metody generyczne (googlujemy).

Mapy:

0. Utwórz i przypisz do zmiennej (capitalByCountry) mapę, która jako klucze i wartości będzie przyjmowała łańcuchy znaków.

Wstaw do mapy pary wartości:

Polska -> Warszawa

Niemcy -> Berlin

Francja -> Paryż

Wypisz w konsoli wszystkie klucze z tej mapy

Wypisz w konsoli wszystkie wartości z tej mapy

1. Tworzymy metodę zadanie1, która za parametr przyjmie listę stringów, a która zwróci w wyniku mapę, której klucze będą indeksami elementów z listy, a wartościami będą elementy z listy.
2. Tworzymy metodę zadanie2, która za parametr przyjmie mapę, której klucze oraz wartości to Stringi, a która zwróci mapę, która dla klucza "klucze" będzie przechowywała listę kluczy, a dla klucza "wartosci" będzie przechowywała listę wartości.
3. Tworzymy metodę zadanie3, która za parametr przyjmie zbiór stringów, a zwróci mapę, której kluczem będzie każdy łańcuch znaków z parametru (zbioru), a wartością będzie odpowiedź na to, ile dany łańcuch zawiera znaków "a".
4. Tworzymy metodę zadanie4, która za parametr przyjmie listę stringów, która zwróci mapę, której kluczem będzie długość łańcucha znaków, a wartością będzie lista stringów, które mają taką długość.
5. Tworzymy metodę, która przyjmie dwa parametry - mapę <String, String> oraz listę <String>. Metoda zwróci odpowiedź, czy dla każdego elementu listy, mapa ma taką właściwość, że element jest albo kluczem, albo wartością (nie jednocześnie kluczem i wartością).
6. Tworzymy metodę, która za parametr przyjmie łańcuch znaków, a która w wyniku zwróci mapę, której kluczem będą litery, a której wartością będzie liczba wystąpień tej litery w zadanym łańcuchu znaków.

Strumienie:

Stwórzmy listę stringów.

```
private static void zadanie1() {  
    List<String> stringList = Arrays.asList("abc", "def");  
    // .....  
}
```

1. Wyfiltrujmy elementy tej listy w taki sposób, żeby w wyniku uzyskać listę stringów, których długość jest większa niż 3.
2. Mapujemy elementy tej listy wybierając ich pierwsze trzy znaki (metoda substring). Następnie zbieramy (collect) dane ze strumienia do listy.
3. Listę stringów przefiltrować tak, żeby zostawić tylko te, które kończą i rozpoczynają się na literę 'A'. Następnie zliczyć je (metoda count())
4. Listę stringów zmapować tak, żeby usunąć ostatnią literę, a następnie przefiltrować tak, żeby zostały tylko teksty, które mają długość przynajmniej 2.

Stworzyć klasę Person, która będzie miała pola:

```
private String name;  
private String surname;  
private int age;  
private float height;  
private String comment;
```



Odwiedź nas

Pomorski Park Naukowo Technologiczny, II piętro  
Al. Zwycięstwa 96/98  
81-451 Gdynia



Zadzwoń do nas

Telefon: 58 782 48 88  
Poniedziałek - Piątek, 8:00 - 16:00



Napisz do nas

Sprawy ogólne: info@sdacademy.pl  
Trenerzy: trenerzy@sdacademy.pl

www.sdacademy.pl



facebook.com/sdacademypl/



oraz konstruktor ustawiający wartości pól oraz gettery i settery.  
Stworzyć listę z elementami typu Person.

5. Z listy osób wypisać imię i nazwisko osób, których wiek jest większy od 20 lub wzrost jest większy niż 1.75.
6. Wyfiltrować osoby tak, żeby zostały osoby o wieku mniejszym niż 18. Następnie przypisać do zmiennej typu List<String> imię oraz nazwiska tych osób.
7. Sprawdzić, czy w liście osób istnieje osoba o nazwisku "Kowalski".
8. Sprawdzić, czy w liście osób wszystkie osoby mają mniej niż 2 metry wzrostu.
9. Posortuj listę osób w następujący sposób - pierw nazwisko alfabetycznie rosnąco, następnie imię.
10. Posortuj listę list liczb całkowitych za pomocą sumy ich elementów.
11. Skonwertować listę do mapy, której kluczem będzie imię i nazwisko, a wartością będzie wiek osoby. (wygooglować sposób konwersji strumienia na mapę).
12. Skonwertować listę do mapy, której kluczem będzie imię, a wartością będzie zbiór osób o tym imieniu.
13. Mamy mapę o strukturze - klucze to stringi, a wartości to listy, których elementy są typu float. Za pomocą strumienia uzyskamy listę kluczy z mapy, w kolejności posortowanej poprzez średnią elementów ich wartości.
14. Stwórz listę list liczb. Wyfiltruj tę listę w taki sposób, aby pozostały jedynie te elementy, które posiadają element o wartości 5.

Optional:

1. Znajdź w kolekcji liczb całkowitych liczbę z przedziału 100 - 115. Przypisz wynik do zmiennej typu Optional.
2. Sprawdź, czy do zmiennej z zadania 1 jest przypisana jakaś wartość, jeśli tak, to ją wypisz, jeśli nie, to wypisz w konsoli "brak wartości z przedziału 100 - 115".
3. W liście stringów wyszukaj jakikolwiek tekst, który rozpoczyna się od "ab". Następnie za pomocą metody ifPresent wypisz w konsoli "Znalazłem tekst: ....", gdzie .... to znaleziony łańcuch znaków.
4. Napisz metodę, która przyjmie za parametr listę osób. Z tej listy osób wyfiltruje osoby - zostawi te, których inicjały to MK. Następnie wybierze wiek najstarszej osoby. Jeśli nie znajdzie takiej osoby, to zwróci -1. Zadanie rozwiąż za pomocą jednej instrukcji.

Wyjątki:

1. Napisz metodę divide, która przyjmie dwa parametry typu int, która w przypadku dzielenia przez 0 zwróci null oraz w konsoli wyświetli odpowiednią informację, zamiast wyrzucania wyjątku (w treści metody używając składni try/catch).
2. Napisz metodę divideTwoArrayElements, która przyjmie trzy parametry: tablicę intów, indeks pierwszego elementu, oraz indeks drugiego elementu, która zwróci wynik dzielenia pierwszego elementu oraz drugiego elementu. Metodę należy napisać w taki sposób, żeby w żadnym wypadku nie wyrzucała wyjątku. Metoda, gdy podano nieprawidłowe argumenty, powinna zwrócić null (w treści metody używając składni try/catch).

-----  
Stworzyć klasę Person:

```
private String name;  
private String surname;  
private int age;  
private float height;  
private String comment;
```

oraz klasę PersonList, która będzie rozszerzała klasę ArrayList<Person>, oraz która będzie miała metody:

3. Metoda wyszukiwająca osoby o zadanym wieku. W przypadku, gdy jako parametr przekaze się wartość ujemną wyrzucimy IllegalArgumentException z informacją, że należy przekazać jako argument wartość większą od 0.



Odwiedź nas

Pomorski Park Naukowo Technologiczny, II piętro  
Al. Zwycięstwa 96/98  
81-451 Gdynia



Zadzwoń do nas

Telefon: 58 782 48 88  
Poniedziałek - Piątek, 8:00 - 16:00



Napisz do nas

Sprawy ogólne: info@sdacademy.pl  
Trenerzy: trenerzy@sdacademy.pl

www.sdacademy.pl



facebook.com/sdacademypl/





4. Metoda wyszukująca jedną osobę o zadanym imieniu i nazwisku. W przypadku, gdy takiej osoby nie będzie na liście, metoda powinna wyrzucić wyjątek typu `NoSuchPersonException` - który będzie miał pola `String name` oraz `String surname`.

Pliki:

1. Stwórz plik `dane.txt`, w którym zapisz kilka linijek tekstem.

Napisz metodę `printFile`, która za parametr przyjmie łańcuch znaków - nazwę pliku, którego wszystkie linie zostaną wypisane. Jeśli plik nie istnieje, to metoda zwróci w wyniku wartość `false`, w innym wypadku zwróci `true`.

2. Stwórz metodę `saveToFile`, która za parametr przyjmie nazwę pliku, listę stringów oraz informację, czy dane pliku chcemy nadpisać.

Po napisaniu metody działającej dla istniejącego pliku, dopisujemy kod, który sprawi, że metoda zadziała również w przypadku, kiedy zadany plik nie istnieje.

3. Tworzymy klasę `Person` o polach opisujących imię, nazwisko, oraz rok urodzenia osoby. W klasie `Person` tworzymy metodę `saveToFile`, która za parametr przyjmie nazwę pliku, a która zadziała tak, że zapisze w pliku oddzielone przecinkami dane osoby.

Tworzymy klasę `Person` z wybranymi polami.

4. W klasie `Person` tworzymy statyczną metodę `getFromFile`, która przyjmie za parametr nazwę pliku, a która zadziała odwrotnie do metody z zadania trzeciego (utworzy i zwróci obiekt typu `Person` zgodnie z danymi z linii pliku).

5. Tworzymy w klasie `Person` metodę statyczną `saveToFile`, która za parametr przyjmie listę obiektów typu `Person` oraz nazwę pliku, a która zapisze w pliku dane dotyczące zadanej listy osób

6. Tworzymy w klasie `Person` metodę statyczną `getListFromFile`, która za parametr przyjmie nazwę pliku, a która zwróci listę osób, zgodnie z danymi z wierszy pliku.

Daty:

1. Napisz funkcję, która przyjmie za parametr łańcuch znaków w formacie `XX.XX.XXXX`, a która zwróci obiekt typu `LocalDate`, o odpowiedniej wartości.

2. Napisz funkcję, analogiczną do tej z zadania 1., która zwróci obiekt typu `LocalTime`

3. Napisz funkcję, która przyjmie za parametry obiekt typu `LocalDate` oraz `LocalTime`, a która zwróci obiekt `LocalDateTime` o odpowiednich wartościach.

4. Posortuj listę dat (`LocalDate`) korzystając ze strumieni.

5. Napisz metodę, która za parametr przyjmie rok (`int`), a która zwróci informację, czy rok jest przestępny.

6. Napisz metodę, która za parametr przyjmie obiekt typu `LocalDate`, a która zwróci datę, będącą poniedziałkiem w tygodniu, w którym znajduje się data przekazana jako parametr.

7. Sprawdź, w którym dniu tygodnia się urodziłeś/aś.

8. Rozwiąż zadanie 5. bez użycia typów złożonych.

Wątki:

1. Napisz program, który jednocześnie będzie niezależnie odliczał od 10 do 1 co pół sekundy i od 1 do 10 co sekundę.

2. Napisz program, w którym użytkownik może dodawać elementy do listy. Program automatycznie co 10 sekund wypisuje zawartość listy.

3. Stwórz projekt reprezentujący wyścig, w którym bierze udział kilku biegaczy. Każdy biegacz może biec z inną prędkością i jest reprezentowany przez inny wątek.

Co 5 sekund wypisywana są pozycje biegaczy z wyścigu (osobny wątek). Dodatkowo co 10 sekund każdy biegacz wypisuje ile przebiegł.



Odwiedź nas

Pomorski Park Naukowo Technologiczny, II piętro  
Al. Zwycięstwa 96/98  
81-451 Gdynia



Zadzwoń do nas

Telefon: 58 782 48 88  
Poniedziałek - Piątek, 8:00 - 16:00



Napisz do nas

Sprawy ogólne: [info@sdacademy.pl](mailto:info@sdacademy.pl)  
Trenerzy: [trenerzy@sdacademy.pl](mailto:trenerzy@sdacademy.pl)

[www.sdacademy.pl](http://www.sdacademy.pl)



[facebook.com/sdacademypl/](https://facebook.com/sdacademypl/)

