

102. W tablicy typu ArrayList zawierającej ciągi znaków zamień wszystkie ciągi na zapisane wielkimi literami. Wykonaj to za pomocą a) iteratora, b) pętli przechodzącej po wartościach indeksu. Na końcu wypisz zawartość tej kolekcji

103. Użyj klasy statycznej Collections i napisz metody które:

- a. Odwróć kolejność elementów w liście
- b. Zmień kolejność elementów na losową
- c. Stwórz kopie listy

104. * Napisz program który będzie symulował rejestrację nowych użytkowników.

- a. Stwórz klasę użytkownik z polami email, imię, nazwisko
- b. Stwórz metodę która będzie dodawała nowego użytkownika ale:
 - i. Nie może istnieć więcej niż 1 użytkownik o tym samym emailu
 - ii. * program powinien wyrzucić wyjątkiem jeżeli dany użytkownik już istnieje
- c. *Stwórz metodę która usunie użytkownika o podanym emailu
- d. Stwórz metodę która wypisze ile jest użytkowników
- e. **Stwórz metodę która wypisze ile jest dziewczyn wśród użytkowników (załóżmy że dziewczyna ma imię kończące się na „a”)

105. W zbiorze typu HashSet zawierającej ciągi znaków zamień wszystkie ciągi na zapisane wielkimi literami. Wykonaj to za pomocą a) iteratora, b) pętli przechodzącej po wartościach indeksu. Na końcu wypisz zawartość tej kolekcji

106. Napisz program wczytujący słowa z pliku i wypisz ile jest unikatowych elementów w tym pliku

107. Napisz mapę która na podstawie wartości liczbowej od 1 do 12 zwróci wartość String z nazwą miesiąca.

108. Napisz program wczytujący wszystkie słowa z pliku i wyświetlający liczbę wystąpień każdego ze słów, słowa niech będą posortowane wg domyślnego sortowania.

109. Napisz swoją implementację klasy Pair która przyjmuje dwa dowolne elementy.

110. Zmień klasę z poprzedniego ćwiczenia poprzez ograniczenie jej argumentów tylko do liczb.

111. * Napisz generyczny interfejs Dao który będzie przyjmował klasy dowolnego typu oraz zawierał takie metody jak:

-save
-update
-delete
-getByName

Następnie napisz jej implementację która będzie operowała na ArrayList

112. Napisz metodę która przyjmie tablice dowolnych elementów (korzystając z metod generycznych) i wypisze wszystkie jej elementy

113. * Napisz metode max która przyjmie 3 dowolne elementy i korzystając z metod CompareTo wyliczy wartość maksymalna i ją zwróci

114. Napisz metodę która przyjmuje listę typu String a następnie sortują te wartości od najdłuższych do najkrótszych słów.

115. Napisz klasę Produkt posiadająca pola nazwa i cena (BigDecimal) następnie napisz domyślny komparator dla tej klasy, który będzie sortował według ceny.

116. Napisz klasę User z polami id (long), email (String) i name (String) a następnie do tej klasy dopisz

- a. Metodę która pobiera nazwę użytkownika a jeżeli jej nie ma to zwróci wartość domyślna „Nieznany użytkownik” (wykorzystaj Optional orElse)
- b. Metodę która pobiera id użytkownika, jeżeli on nie istnieje to zwróci wyjątek z treścią „nie może istnieć użytkownik bez identyfikatora” (wykorzystaj Optional orElseThrow)

117. Napisz program, który pobierze o użytkownika cztery łańcuchy znaków, które umieścisz w liście. Następnie posortuj tę listę używając metody sort. Użyj wyrażenia lambda, które posortuje łańcuchy znaków malejąco po długości.

118. Napisz metodę która przyjmuje listę zawierająca Stringi a następnie zamieni wszystkie te Stringi na napisane wielkimi literami i zwróci tylko unikalne rekordy

119. Napisz klasę Produkt posiadająca pola nazwa i cena (w wersji łatwiejszej dla ceny daj typ Double a w wersji trudniejszej użyj BigDecimal) następnie stwórz listę takich produktów i napisz z wykorzystaniem Streamów:

- a. Metodę która wyznaczy produkt z najniższą ceną
- b. Metodę która wyznaczy ile łącznie kosztują dane produkty
- c. Metodę która zwróci listę produktów których nazwa zaczyna się na „a”
- d. Metodę która zwróci listę produktów których cena jest pomiędzy 10 a 20
- e. Metodę która pogrupuje produkty według ceny

120. Napisz program, który utworzy 10 wątków. Zadbaj o to, by każdy z nich znał swój numer (0..9). W treści wątku wypisuj w pętli liczby o wartości $i * 10 + N$, gdzie N jest numerem wątku;

121. Napisz program, w którym grupa wątków określoną liczbę razy będzie zwiększała wartość dzielonej przez nie zmiennej. W wątku głównym za pomocą metody join() zaczekaj

na zakończenie wątków zwiększających i wypisz końcową wartość zmiennej.

Uzupełnij program o synchronizację wątków gwarantującą wyłączenie wykonania kodu zwiększającego wartość zmiennej.

122. Zmodyfikuj rozwiązanie ćwiczenia poprzedniego, zastępując jawną synchronizację wątków użyciem zmiennej atomowej.

123. Zmodyfikuj rozwiązanie z zadania poprzedniego by używało `ExecutorService`