

### Zadanie 1

Utwórz dowolną klasę, a potem zainicjalizowaną tablicę obiektów tej klasy. Zawartością tablicy wypełnij listę List. Wyłuskaj z niej fragment listy metodą subList(), a następnie usuń tę podlistę z oryginalnej listy.

### Zadanie 2

Utwórz i zapełnij listę List<Integer>. Utwórz drugą listę List<Integer>. Użyj ListIterator do przejrzenia elementów pierwszej listy i wstawienia ich do listy drugiej, ale w odwrotnej kolejności.

### Zadanie 3

Napisz klasę o nazwie Command, która zawiera ciąg znaków String i metodę operation(), która go wypisuje. Napisz drugą klasę, z metodą wypełniającą kolejkę Queue obiektami klasy Command i zwracającą wypełniony kontener. Przekaż kontener do metody z trzeciej klasy: metoda ma konsumować obiekty z kolejki Queue, wywołując dla każdego z nich metodę operation().

### Zadanie 4

1. Utwórz nową klasę Kangur ze składową int nrKangura, inicjalizowaną z poziomu konstruktora. Wyposaź klasę w metodę skok(), wypisującą wartość tej składowej i sygnalizującą wykonywanie podskoków. Utwórz kontener ArrayList i wstaw do niego obiekty Kangur (minimum 10). Teraz skorzystaj z metody get() kontenera w celu przejrzenia jego zawartości i wywołania metody skok() dla każdego umieszczonego w nim kangura.
2. Zmodyfikuj uzyskany kod tak, aby przeglądało listę (i wywoływało metodę skok()) za pomocą iteratora.
3. Weź klasę Kangur z podpunktu 1 i umieść jej elementy w kontenerze HashMap, kojarząc każdy egzemplarz Kangur (wartość) z nazwą ("Jacek", "Marta" itd.) w postaci obiektu String (klucz). Pozyskaj iterator zbioru zwracanego przez keySet() i wykorzystaj go do przejrzenia kontenera HashMap. Wypisz w konsoli imiona kangurów oraz odpowiadające im numery, zwracane poprzez metodę skok().
4. Wyodrębnij z kontenera HashMap (utworzonego w poprzednim podpunkcie) pary, posortuj je według kluczy i umieść całość w kontenerze LinkedHashMap.

### Zadanie 5

W poniższym zadaniu napiszemy własną klasę odnośnie stosu, czyli co zostanie włożone na stos jako ostatnie jest pierwszym elementem, który można z niego zdjąć (LIFO).

1. Utwórz klasę Stos<T> zawierającą:
  - zainicjowane prywatne pole LinkedList<T> stos,
  - publiczne metody:
    - void push(T v) - wkłada element na stos,
    - T peek() - zwraca pierwszy element stosu, ale go nie usuwa,
    - T pop() - zwraca pierwszy element stosu i usuwa go,
    - boolean empty() - sprawdza, czy stos jest pusty,
    - String toString() - wypisuje elementy naszego stosu.

2. Stosy są często używane do obliczania wyrażeń w językach programowania. Za pomocą utworzonej klasy Stos oblicz poniższe wyrażenie, w którym '+' oznacza "umieszczenie następnej litery na stosie", a '-' "zjęcie szczytowego elementu stosu i wypisanie go na wyjściu".

Wyrażenie do wyliczenia: "+B+a+l---+a+g+a---+n--+w--+l+i+t---+e--+r+k--+a+c+h---"

### **Zadanie 6**

Napisz prostą symulację pójścia na zakupy:

1. utwórz tablicę String zawierającą 10 nazw produktów dostępnych w sklepie,
2. po wejściu do sklepu zapełnij koszyk (będący stosem - użyj klasy z poprzedniego zadania) losową ilością losowych towarów (koszyk pomieści maksymalnie 15 przedmiotów),
3. umieść siebie w kolejce (Queue) na losową pozycję (maksymalna ilość osób w kolejce wynosi 10),
4. gdy dojdiesz do kasy wypisz produkty znajdujące się w koszyku.

### **Zadanie 7**

1. Napisz klasę Film zawierającą prywatne pole czasTrwania, tytuł oraz czyObejrzany (wartość true gdy film został obejrzany). Dodaj konstruktor i odpowiednie metody obsługujące pola.
2. Stwórz dwie klasy dziedziczące z klasy Film, np. Horror i Komedia. Każda z nich ma zawierać pole finalne pole typ (wskazujące na typ filmu). Dodaj konstruktory i odpowiednie metody obsługujące pola. Dodaj metody wypisujące wszystkie dane odnośnie filmu.
3. W głównej klasie stwórz mapę prywatnaKolekcja, gdzie kluczem będzie numer egzemplarza, a wartością obiekt Film. Dodaj kilka elementów do utworzonej mapy.
4. Za pomocą iteratora wypisz posiadane filmy w kolekcji. Przykład:  
"Nr (nr w kolekcji), Tytuł: (tytuł filmu), Czas trwania: (czas), Obejrzany: (tak/nie), Typ: (gatunek filmu)"