```
import java.util.ArrayList;
import java.util.Collections;

public class ArrayListDemo {
    public static void main(String[] args) {

        // ------------------------------------------------------------
        // 1. TWORZENIE ARRAYLIST
        // ------------------------------------------------------------
        // Tworzymy nową, pustą listę przechowującą elementy typu String.
        // ArrayList jest dynamiczną tablicą – może się rozszerzać podczas działania programu.
        ArrayList<String> fruits = new ArrayList<>();

        // Wyświetlenie pustej listy
        System.out.println("Pusta lista: " + fruits);



        // ------------------------------------------------------------
        // 2. DODAWANIE ELEMENTÓW DO LISTY
        // ------------------------------------------------------------
        // add(element) - dodaje element na końcu listy
        fruits.add("Jabłko");
        fruits.add("Banan");
        fruits.add("Pomarańcza");

        // add(index, element) - dodaje element na konkretnym indeksie, przesuwając
pozostałe elementy
        fruits.add(1, "Gruszka");

        System.out.println("\nLista po dodaniu elementów: " + fruits);



        // ------------------------------------------------------------
        // 3. POBIERANIE ELEMENTÓW
        // ------------------------------------------------------------
        // get(index) - pobiera element z podanej pozycji (indeks od 0!)
        String firstFruit = fruits.get(0);
        String secondFruit = fruits.get(1);

        System.out.println("\nPierwszy owoc: " + firstFruit);
        System.out.println("Drugi owoc: " + secondFruit);

        // Jeśli użyjemy get() z indeksem poza zakresem — pojawi się
IndexOutOfBoundsException
```

```
// ------------------------------------------------------------
// 4. USUWANIE ELEMENTÓW
// ------------------------------------------------------------
// remove("wartość") – usuwa pierwsze wystąpienie elementu o tej wartości
fruits.remove("Banan");

// remove(index) – usuwa element o podanym indeksie
fruits.remove(0);

System.out.println("\nLista po usunięciu elementów: " + fruits);



// ------------------------------------------------------------
// 5. ITEROWANIE PO LIŚCIE
// ------------------------------------------------------------

// Najprostszy sposób — pętla for-each
System.out.println("\nIteracja po liście (for-each):");
for (String fruit : fruits) {
    System.out.println(fruit);
}

// Iteracja z użyciem indeksów — pozwala np. modyfikować elementy
System.out.println("\nIteracja po liście (for z indeksem):");
for (int i = 0; i < fruits.size(); i++) {
    System.out.println(fruits.get(i));
}



// ------------------------------------------------------------
// 6. INFORMACJE O LIŚCIE
// ------------------------------------------------------------
// size() – liczba elementów w liście
System.out.println("\nRozmiar listy: " + fruits.size());

// isEmpty() – sprawdza, czy lista jest pusta
System.out.println("Czy lista jest pusta? " + fruits.isEmpty());

// contains(element) – sprawdza, czy lista zawiera dany element
System.out.println("Czy lista zawiera 'Pomarańcza'? " + fruits.contains("Pomarańcza"));

// indexOf(element) – zwraca indeks pierwszego wystąpienia lub -1 gdy brak
System.out.println("Indeks 'Pomarańcza': " + fruits.indexOf("Pomarańcza"));
System.out.println("Indeks 'Banan': " + fruits.indexOf("Banan")); // -1 bo "Banan" został
wcześniej usunięty
```

```
// ------------------------------------------------------------
// 7. MODYFIKACJA ELEMENTÓW
// ------------------------------------------------------------
// set(index, element) – zamienia element znajdujący się na danym indeksie
fruits.set(1, "Kiwi");
System.out.println("\nLista po zmianie elementu: " + fruits);



// ------------------------------------------------------------
// 8. KOPIOWANIE LISTY I CZYSZCZENIE
// ------------------------------------------------------------
// Tworzenie kopii listy — użycie konstruktora ArrayList
ArrayList<String> copyFruits = new ArrayList<>(fruits);

System.out.println("\nKopia listy: " + copyFruits);

// clear() – usuwa wszystkie elementy z listy
fruits.clear();
System.out.println("Lista po wyczyszczeniu: " + fruits);



// ------------------------------------------------------------
// 9. SORTOWANIE LISTY
// ------------------------------------------------------------
// sort() – sortuje elementy w porządku alfabetycznym (dla String)
Collections.sort(copyFruits);
System.out.println("\nPosortowana kopia listy: " + copyFruits);

// reverse() – odwraca kolejność elementów
Collections.reverse(copyFruits);
System.out.println("Lista po odwróceniu: " + copyFruits);



// ------------------------------------------------------------
// 10. ARRAYLIST Z LICZBAMI
// ------------------------------------------------------------
// Tworzenie listy z liczbami typu Integer (typ prosty int nie jest dozwolony!)
ArrayList<Integer> numbers = new ArrayList<>();
numbers.add(5);
numbers.add(10);
numbers.add(3);
```

```
        System.out.println("\nLista liczb: " + numbers);

        // sort() – sortuje liczby rosnąco
        Collections.sort(numbers);
        System.out.println("Posortowane liczby: " + numbers);

        // reverse() – odwraca kolejność liczb
        Collections.reverse(numbers);
        System.out.println("Liczby po odwróceniu: " + numbers);
    }
}
```