

Napisy

```
String s = "alamakota";
int slen = s.length(); - długość napisu
String su = s.toUpperCase();
String sr = s.replace("ala", "ola");
boolean sb = s.contains("kota"); - czy zawiera dany wzorzec
String ss = s.substring(1, 3); - podslowo
```

Tablice

```
int a[] = new int[]{1, 2, 4, 8, 16};
Arrays.sort(a); - sortowanie tablicy
int idx1 = Arrays.binarySearch(a, 4); - wyszukiwanie binarne
konwersja do listy:
List<Integer> l = IntStream.of(a).boxed().collect(Collectors.toList());;
```

Kolekcje

```
ArrayList<String> arr = new ArrayList<>();
arr.add("X"); - dodawanie pojedynczego elementu
dodawanie wielu elementów::
Collections.addAll(arr, new String[]{"Y", "Z"});
arr.remove("Y"); - usunięcie pojedynczego elementu
String s0 = arr.get(0); - dostęp do pierwszego elementu
arr.set(0, "XX"); - zastąpienie pierwszego elementu
Collections.sort(arr); - sortowanie
Collections.sort(arr, Collections.reverseOrder()); - odwrotne
pętle po wszystkich elementach:
for(String x: arr) System.out.println(x);
arr.forEach(x -> System.out.println(x));
arr.forEach(System.out::println);
map:
arr.stream().map(x -> x + x).forEach(x -> System.out.println(x));
filtrowanie:
arr.stream().filter(x -> x.startsWith("X")).findFirst().ifPresent(System.out::println);
grupowanie:
Map<String, List<String>> g = arr.stream().collect(Collectors.groupingBy(x -> x.substring(0, 1)));;
```

Słowniki

```
TreeMap<String, Integer> d = new TreeMap<>();
d.put("X", 1); - dodawanie nowej pary (klucz, wartość)
int dx = d.get("X"); - pobieranie wartości
int dy = d.getOrDefault("Y", -1); - pobieranie wartości (o ile istnieje)
boolean b = d.containsKey("X"); - sprawdzanie czy zawiera klucz
```

Wczytywanie danych

```
Scanner sc = new Scanner(input);
int x1 = sc.nextInt(); - wczytaj następną liczbę
String w = sc.next(); - wczytaj następny token
```

Liczby losowe

```
Random r = new Random();
int x = r.nextInt(2); - losowa liczba całkowita z zakresu [0, bound)
float y = r.nextFloat(); - losowa liczba z zakresu [0, 1]
```