

Zadania

Zadanie 1

Czy *List* to interfejs czy klasa?

Zadanie 2

Czy *ArrayList* to interfejs czy klasa?

Zadanie 3

Czy *LinkedList* to interfejs czy klasa?

Zadanie 4

Czym się różni *ArrayList* od *LinkedList*?

Zadanie 5

Co wydrukuje poniższy kod?

```
public class Main {  
  
    public static void main(String args[]) {  
  
        ArrayList<Lato> al = new ArrayList<Lato>();  
        al.add(new Lato());  
        System.out.println(al.get(0).a.length);  
  
    }  
  
}  
  
class Lato {  
    String[] a = {"a","b"};  
}
```

Zadanie 6

Popraw poniższy kod, tak by kompilator przestał zgłaszać problemy.

```
String[] s = {"a","b"};  
  
ArrayList<String> al = new ArrayList<>();  
al.add(s);
```

Zadanie 7

Dołącz wszystkie elementy z listy składki2 do listy składki.

a) umieść elementy składki2 na końcu listy składki,

b) umieść elementy składki 2 na początku listy składki

```
ArrayList<Double> skladki = new ArrayList<Double>();
skladki.add(3.14);
skladki.add(1.414);
skladki.add(2.718);

ArrayList<Double> skladki2 = new ArrayList<Double>();
skladki2.add(1.73);
skladki2.add(2.23);
```

Zadanie 8

Zmień poniższy kod, tak by się zaczął kompilować. Spróbuj dodać to listy *ptaki* jakieś ptactwo.

```
public class Main {

    public static void main(String args[]) {

        ArrayList<Ptak> ptaki = new ArrayList<>();

    }

}

class Kura {

}

class Kaczka {

}
```

Zadanie 9

Usuń pierwszą pozycję z listy.

```
ArrayList<Integer> arr = new ArrayList<>();
arr.add(1);
arr.add(2);
arr.add(3);
```

Zadanie 10

Podnieś wszystkie elementy do potęgi. Możesz użyć np. `Math.pow(...)`.

```
ArrayList<Double> arr = new ArrayList<>();
arr.add(1.0);
arr.add(2.0);
arr.add(3.0);
```

Zadanie 11

Czy poniższa próba uruchomienia kodu się powiedzie?

```
List<Integer> al = new ArrayList<>();  
al.add(null);
```

Zadanie 12

Czy List jest uporządkowaną czy nieuporządkowaną kolekcją?

Zadanie 13

Czy List może zawierać duplikaty?