

# Programowanie obiektowe w Javie (POJ)

## Laboratorium 9

Każde z poniższych zadań wykonaj w osobnym projekcie/ewentualnie pakiecie wewnątrz jednego projektu. Pamiętaj o właściwym stosowaniu modyfikatorów dostępu i enkapsulacji detali implementacji, które nie powinny być widoczne dla innych klas.

### Zadanie 1.

Pobierz szkielet klasy Lista z folderu 'Materiały do zadań'. Twoim zadaniem jest zaimplementowanie pozostałych metod opisanych poniżej:

- dodaj(E element):boolean – powinna umożliwiać dodawanie kolejnego elementu, w sytuacji gdy lista jest pełna powinna zwrócić false i nie dodać elementu
- czyZawiera(E element):boolean – zwraca informacje o tym czy element znajduje się wewnątrz listy
- zwrocIndeks(E element):int – zwraca numer indeksu pierwszego wystąpienia wskazanego elementu. W przypadku braku elementu niech zwraca -1
- zwrocOstatniIndeks(E element):int – zwraca numer indeksu ostatniego wystąpienia danego elementu. W przypadku braku elementu niech zwraca -1
- zwrocElement(int indeks):E – zwraca element znajdujący się na wskazanym indeksie. Jeśli nie ma elementu niech zwróci null lub wyjątek własny (nie pozwolić na ArraysOutOfBoundsException)
- podmienElement(int indeks, E element):boolean – podmienia element znajdujący się pod danym indeksem. Jeśli wskażemy niewłaściwy indeks lub nie uda się podmienić elementu metoda powinna zwrócić false

Następnie przetestuj swoje rozwiązanie dla 2 wybranych typów danych np. String, Integer, Osoba itp.

### Zadanie 2.

Pobierz szkielet klasy Słownik z folderu 'Materiały do zadań'. Twoim zadaniem jest zaimplementowanie pozostałych metod opisanych poniżej:

- dodaj(K klucz, V wartość):boolean – dodaje nową parę klucz-wartość, zwraca false jeśli wielkość przekroczy rozmiar słownika. Jeśli istnieje już wskazany klucz to metoda powinna nadpisać wartość zgodnie z przekazanym parametrem
- czyZawiera(K klucz):boolean – zwraca true jeśli słownik zawiera dany klucz
- wielkość():int – zwraca ilość istniejących haseł słownika (nie mylić z rozmiarem w konstruktorze)

Następnie utwórz 2 zestawy kluczy i wartości i sprawdź działanie metod oraz wyświetl pary kluczy i wartości np. Integer->String, String->String, String->Osoba itp.

### Zadanie 3.

Zaimplementuj metodę *wczytajSłownik(File file)* w klasie Słownik, tak aby wczytywała dane z pliku tekstowego np. tłumaczenia Pol->Ang. Wybierz dowolny dla siebie sposób rozróżniania haseł. Po wykonaniu tej metody powinieneś otrzymać obiekt typu Słownik o wielkości haseł zdefiniowanych w pliku tekstowym.

### Zadanie 4.

Utwórz klasę Zawodnik o polach : imie, nazwisko, numer, predkosc max. Dodaj odpowiedni konstruktor oraz gettery, a następnie:

- Utwórz klasę ImieComparator

- b) Zaimplementuj interfejs Comparator
- c) Niech umożliwi sortowanie po imieniu
- d) Powtórz kroki a-c dla numeru i prędkości max
- e) Utwórz kilku zawodników i posortuj po każdym rodzaju comparatora

*Rozwiązanie zadań powinno być udostępnione w na indywidualnym repozytorium github studenta maksymalnie do końca czwartego dnia po zajęciach:*

*- zajęcia w poniedziałek -> termin do piątku 23:59:59*

*- zajęcia w środę -> termin do niedzieli 23:59:59*

*- zajęcia w czwartek -> termin do poniedziałku 23:59:59*

*Zadania powinny być zrealizowane w osobnych klasach w ramach jednego projektu, a umieszczone w prywatnym repozytorium, które jest udostępnione prowadzącemu (<https://github.com/plucins>)*