

## Laboratorium 4

Jednym z ważnych problemów w programowaniu jest umiejętność doboru odpowiedniej struktury danych. Java w tym obszarze dostarcza szereg rozwiązań, a celem zajęć jest analiza właściwości poszczególnych z nich:

### Zad 1

Zadanie polega na analizie złożoności obliczeniowej różnych implementacji klas Lista. W tym celu

1. Stwórz pakiet <grupa>.pu.lab4 gdzie <grupa> to identyfikator grupy np. BSI1
2. Stwórz klasę *TestList* zawierającą metodę startową *main()*, metoda ta powinna realizować kolejne testy operacji na Listach (dodawanie, odczyt, usuwanie itp.) oraz mierzyć czasy ich wykonania. Pomiar czasu wykonania realizowany jest poprzez dokładny pomiar czasu przed wykonaniem operacji i dokładny pomiar po wykonaniu operacji. Pomiar czasy realizowany jest za pomocą

```
long czas = System.currentTimeMillis()
```

3. Dodaj do klasy metody o sygnaturze:

```
public static void listRemoveRnd(List<Number> list, Random random) {...5 lines }
public static void listRemoveFirst(List<Number> list) {...8 lines }
public static void listRemoveLast(List<Number> list) {...5 lines }
public static void listAddTest(List<Number> list, int iterations, Random random) {...6
public static int listRndReadTest(List<Number> list, Random random) {...10 lines }
public static int listSeqForReadTest(List<Number> list) {...9 lines }
public static int listSeqForEachReadTest(List<Number> list) {...8 lines }
```

Metoda:

- *listAddTest* powinna uzupełniać listę wartościami losowymi. Do listy powinno zostać dodanych iterations wartości
- *listRndReadTest* – powinna odczytać z listy w sposób losowy n wartości, gdzie n to liczba elementów na liście. Odczytane wartości powinny zostać ze sobą dodane
- *listSeqForReadTest* – w metodzie powinny zostać odczytane wartości w sposób sekwencyjny jedna po drugiej za pomocą zwykłej pętli `for(int i=1; i<s; i++)`. Metoda powinna zwracać sumę wartości odczytanych liczb
- *listSeqForEachReadTest* – Wynik działania powinien być identyczny z powyższą funkcją, z tym że liczby z listy powinny zostać odczytane za pomocą pętli `for-each`
- *listRemoveFirst* – usunięcie z listy wszystkich elementów, w ten sposób iż usuwany jest zawsze pierwszy element
- *listRemoveLast* – z listy powinny zostać usunięte wszystkie elementy w ten sposób że zawsze usuwany jest ostatni element
- *listRemoveRnd* – z listy powinny zostać usunięte wszystkie elementy ale tak, iż usuwane są one w sposób losowy.

Uwagi: pamiętaj iż każdorazowo przed usunięciem elementów z listy należy listę uzupełnić wywołując metodę *listAddTest*

## Zad 2

Badanie struktury typu Mapa