**Выводы.**

В ходе работы были проведены сравнения различных коллекций по времени исполнения основных операций.

**Пункт А.**

Сравнение ArrayList и LinkedList. **Операции для сравнения**: добавление элемента, поиск элемента по значению, удаление элемента. Результаты измерения представлены в миллисекундах в таблице ниже:

|  |  |  |
| --- | --- | --- |
|  | ArrayList | LinkedList |
| add() | 6 | 5 |
| IndexOf() | 7 | 16 |
| Remove() | 12 | 11 |

На основании полученных данных можно сделать выводы, что при добавлении элемента лучше себя показывает LinkedList, при поиске элемента – ArrayList, и при удалении элемента – LinkedList.

**Пункт Б.**

Сравнение HashSet, LinkedHashSet и TreeSet. **Операции для сравнения**: добавление элемента, проверка, содержится ли элемент в коллекции, удаление элемента. Результаты измерения представлены в наносекундах в таблице ниже:

|  |  |  |  |
| --- | --- | --- | --- |
|  | HashSet | LinkedHashSet | TreeSet |
| Add() | 4200 | 700 | 111800 |
| Contains() | 15400 | 4099 | 38700 |
| Remove() | 42400 | 9899 | 19800 |

Во всех трех операциях многим лучше остальных себя показал LinkedHashSet.

**Пункт В.**

Сравнение HashMap, LinkedHashMap и TreeMap. **Операции для сравнения**: добавление элемента, проверка, содержится ли элемент в коллекции, удаление элемента. Результаты измерения представлены в наносекундах в таблице ниже:

|  |  |  |  |
| --- | --- | --- | --- |
|  | HashMap | LinkedHashMap | TreeMap |
| Put() | 77999 | 30800 | 81999 |
| Contains() | 228600 | 111201 | 379500 |
| Remove() | 21000 | 12200 | 97499 |

Как и в предыдущем случае, во всех трех операциях многим лучше остальных себя показала LinkedHashMap.