# КОЛЛЕКЦИИ

# Что такое дженерики.

# Какую проблему они решают.

# Что было до дженериков.

# Что можно типизировать.

# Что такое стирание и сырые типы (raw type - необработанный тип).

# К чему приводит использование raw type.

# Какой механизм обеспечивает обратную совместимость сырых типов и дженериков.

# Если поле типизировано дженериком как в байт коде будет представлен этот тип.

# Как параметризовать статический метод.

# Что такое даймонд оператор.

# Чему эквивалентно <?>.

# Можно ли объявить так: class Animal<?> {}

# Принцип PECS.

# WILDCARD

# Почему последняя строчка не скомпилируется?

**List<ArrayList> arrayLists = new ArrayList<ArrayList>();**

**ArrayList<List> arrayList = new ArrayList<ArrayList>();**

Коллекции.

# Что такое коллекция.

# Какие есть типы коллекций? Как они характеризуются?

# Отличие коллекции от массива.

# Иерархия коллекций.

# 

# Внутреннее устройство коллекций.

# Какие структуры данных вы знаете.

# Как задается порядок следования объектов в коллекции, как отсортировать коллекцию?

# Назовите основные реализации List, Set, Queue, Map.

# Может ли коллекция хранить примитивы? Как можно реализовать хранение примитивов?

List

# Отличие List от Set.

# Как устроен LinkedList.

# Отличие ArrayList и LinkedList.

# Когда лучше использовать ArrayList, а когда LinkedList.

# Скорость вставки элемента в начало середину и конец у ArrayList и LinkedList.

# Как работает метод contains в ArrayList и LinkedList.

# Отличие двусвязного и односвязного списка.

Set

# 3 реализации Set. Какая упорядоченность в какой и почему.

# Как работает HashSet. ()

# Внутреннее устройство HashSet.

# Что кладется на место значения в HashSet.

# Почему в HashSet вместо value не null а new Object.

# Null в TreeSet.

# Как работает метод contains в HashSet.

# Назовите различия между Set и List?

Map

# Что такое Map.

# Что должно быть уникальным в Map.

# Почему Map не входит в Collection.

# Строение HashMap? Что внутри статического класса Node?

# Как работает HashMap.

# Как расширяется HashMap.

# Может ли null быть ключём в HashMap.

# Какой хэш-код у null в HashMap.

# Расскажите подробно, как работает метод put.

# Как работает метод get и remove в HashMap.

# Красно-черное дерево

Чего не гарантирует и что гарантирует красно-черное дерево?

# Что такое бинарное дерево поиска (BST)

# Что такое коллизия. Что происходит при коллизии. Как будет разрешаться коллизия.

# Как перебрать все ключи, значения, пары Map учитывая, что Map - это не Iterable?

Queue

# Что такое Queue.

# В чём разница между Queue и Deque и Stack.

# Что такое Dequeue?

Iterator.

# Что такое Iterator?

# В каких случаях нужно использовать iterator. почему.

# Зачем в итераторе метод remove.

# В чём разница между Iterable и Iterator.

# Как между собой связаны Iterable, Iterator и «forEach»?

# listIterator - что это, в чём отличие от обычного.

# Что произойдет при вызове Iterator.next() без предварительного вызова Iterator.hasNext()?

# Сколько элементов будет пропущено, если Iterator.next() будет вызван после 10-ти вызовов Iterator.hasNext()?