

# Семинар №4

## Задачи семинара

1. Вопросы и обсуждение семинара №3 ( + @Override, @ SuppressWarnings)
2. Использование generic в Java
3. Базовые контейнерные классы: List, Set и Map и их стандартные реализации: ArrayList, LinkedList, HashMap, HashSet, TreeMap.
4. Роль функций Object.hashCode() и Object.equals() при работе с контейнерами
5. Итераторы и оператор foreach
6. Обзор классов java.util.Collections и java.util.Arrays

## Материалы

### Пример использования Generics

```
public class MaxFinder<T extends Comparable> {
    private T max;

    private void add(T num) {
        if (max == null || num.compareTo(max) > 0) {
            max = num;
        }
    }

    private T getMax() {
        return max;
    }

    // for testing
    public static void main(String args[]) {
        MaxFinder<Integer> finder = new MaxFinder<Integer>();
        finder.add(5);
        finder.add(10);
        System.out.println(finder.getMax());
    }
}
```

### List, Set и Map

Основные примитивы:

1. Collection - упорядоченный список элементов, доступ только через итератор
2. List – упорядоченный список элементов с возможностью доступа по индексу
3. Set – множество без дубликатов и без доступа по индексу
4. Map – множество пар ключ-значение, где ключи не повторяются. Нет доступа по индексу

Их основные реализации:

1. ArrayList — аналог массива переменного размера, очень быстрый доступ по индексу
2. LinkedList — список элементов связанных друг с другом ссылками на соседей. Очень медленный доступ по индексу, быстро только первый и последний элемент.
3. HashMap, HashSet — реализации на основе функции hashCode
4. TreeMap, TreeSet – элементы (ключи) должны быть Comparable, реализация на основе дерева. Добавляют понятие упорядоченности. Лучшее время поиска, но большее накладные расходы на построение чем на основе функции hashCode

## Object.hashCode() и Object.equals()

Всегда перегружаются вместе и есть неявное требование что если два объекта равны, то их hash code тоже равен

## Примеры обхода коллекций

### ArrayList + Iterator

```
for(Iterator it = arrayList.iterator(); it.hasNext();){  
    Object cur = it.next();  
    //may perform it.remove():  
}
```

### Map + foreach

```
for(Map.Entry<String, String> entry : someMap){  
    entry.getKey();  
    entry.getValue();  
}
```

## java.util.Collections и java.util.Arrays

Вспомогательные классы реализующие:

1. Wrapper's поверх обычных классов коллекций:
  1. synchronizedMap, synchronizedList, ...
  2. unmodifiable view
  3. Empty Set, List, ...
2. Утилитарные методы для реализации частых алгоритмических задач:
  1. Быстрая сортировка
  2. Бинарный поиск
  3. Изменение порядка на обратный
  4. Поиск max, min
  5. ....

## Open Source Collections Libraries in Java

<http://java-source.net/open-source/collection-libraries>

Основные направления расширения:

- Реализация коллекций для примитивных типов (например Commons Primitives, Trove)
- Расширение возможностей стандартных коллекций (например Google Collections, Commons Collections, Trove)

Commons-Collections seek to build upon the JDK classes by providing new interfaces, implementations and utilities.

There are many features, including:

- Bag interface for collections that have a number of copies of each object
- Buffer interface for collections that have a well defined removal order, like FIFOs
- BidiMap interface for maps that can be looked up from value to key as well and key to value
- MapIterator interface to provide simple and quick iteration over maps
- Type checking decorators to ensure that only instances of a certain type can be added
- Transforming decorators that alter each object as it is added to the collection
- Composite collections that make multiple collections look like one
- Ordered maps and sets that retain the order elements are added in, including an LRU based map
- Identity map that compares objects based on their identity (==) instead of the equals method
- Reference map that allows keys and/or values to be garbage collected under close control
- Many comparator implementations
- Many iterator implementations
- Adapter classes from array and enumerations to collections
- Utilities to test or create typical set-theory properties of collections such as union, intersection, and closure

## Java — задания

Написать программу, которая будет принимать в качестве аргумента имя текстового файла и выводить CSV файл с колонками:

1. Слово
2. Частота
3. Частота%

CSV файл должен быть упорядочен по убыванию частоты, то есть самые частые слова должны идти в начале. Для чтения использовать: `java.io.InputStreamReader`, например:

```
Reader r = new InputStreamReader(new BufferedInputStream(new FileInputStream("имя файла")));
```

Использовать `StringBuilder` класс для построения слова. Разделителями считать все символы кроме букв и цифр (использовать метод `Character.isLetterOrDigit`). Для хранения в памяти можно использовать `Map`, который должен будет хранить объекты специального класса (например `WordCounter`).

Для сортировки (после заполнения) можно создать `List` и скопировать в него `map.values`. Класс `WordCounter` должен реализовать интерфейс `Comparable`.