

## Задание 1. Классы коллекций

	Ordering	Random access	Key-value pairs	Allows duplicates	Allows null values	Thread safe	Blocking operation
ArrayList	Yes	Yes	No	Yes	Yes	No	No
LinkedList	Yes	No	No	Yes	Yes	No	No
Vector	Yes	Yes	No	Yes	Yes	Yes	Yes
Stack	Yes	Yes	No	Yes	Yes	Yes	Yes
CopyOnWriteArrayList	Yes	Yes	No	Yes	Yes	Yes	Yes
LinkedBlockingQueue	No	No	No	Yes	No	Yes	Yes
ArrayBlockingQueue	No	No	No	Yes	No	Yes	Yes
DelayQueue	No	No	No	Yes	No	Yes	Yes
SynchronousQueue	No	No	No	Yes	No	Yes	Yes
LinkedTransferQueue	No	No	No	Yes	No	Yes	No
ArrayDeque	No	No	No	Yes	No	No	No
PriorityQueue	Yes	No	No	Yes	No	No	No
PriorityBlockingQueue	Yes	No	No	Yes	No	Yes	Yes
HashSet	No	No	No	No	Yes	No	No
LinkedHashSet	Yes	No	No	No	Yes	No	No
TreeSet	Yes	No	No	No	No	No	No
EnumSet	Yes	No	No	No	No	No	No
CopyOnWriteArraySet	Yes	Yes	No	No	Yes	Yes	No
HashMap	No	Yes	Yes	No	Yes	No	No
TreeMap	Yes	Yes	Yes	No	Yes	No	No
LinkedHashMap	No	Yes	Yes	No	Yes	No	No
EnumMap	Yes	Yes	Yes	No	No	No	No
WeakHashMap	No	Yes	Yes	No	Yes	No	No
IdentityHashMap	No	Yes	Yes	No	Yes	No	No
ConcurrentHashMap	No	Yes	Yes	No	No	Yes	No

## Задание 3. Ссылки на коллекции.

Определена иерархия классов:

```
class MedicalStaff {}  
class Doctor extends MedicalStaff{}  
class Nurse extends MedicalStaff{}  
class HeadDoctor extends Doctor{}
```

```
Doctor doctor1 = new Doctor() – correct;  
Doctor doctor2 = new MedicalStaff() – not correct;  
Doctor doctor3 = new HeadDoctor() – correct;  
Object object1 = new HeadDoctor() – correct;  
HeadDoctor doctor5 = new Object() – not correct;  
Doctor doctor6 = new Nurse() – not correct;
```

Nurse nurse = new Doctor() – not correct;  
 Object object2 = new Nurse() - correct;  
 List<Doctor> list1 = new ArrayList<Doctor>() - correct;  
 List<MedicalStaff> list2 = new ArrayList<Doctor>() – not correct;  
 List<Doctor> list3 = new ArrayList<MedicalStaff>() – not correct;  
 List<Object> list4 = new ArrayList<Doctor>() - not correct;  
 List<Object> list5 = new ArrayList<Object>() - correct;

## Задание 4. Применение коллекций

	Основная функциональность	Примеры типичного использования
List	Хранение массивов данных	
Set	Хранение множество элементов среди которых исключены повторения	Хранение элементов среди которых нужно избежать повторений или же нет необходимости хранить больше одного элемента с одинаковым значением
Map	Хранение множества пар ключ-значений	Создание словарей, хранение информации из properties файла, хранение элементов и количества их вхождений в каком-то тексте
Queue	Хранение данных и обработка согласно принципу FIFO/LIFO	Очередь номерков, верификация скобочной последовательности