

Задание 1. Классы коллекций

Изучите классы реализации коллекций и заполните следующую таблицу

	Ordering	Random Access	Key-Value Pairs	Allows Duplicates	Allows Null Values	Thread Safe	Blocking Operations
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
PriorityQueue	Yes	Yes	No	Yes	No	No	No
HashSet	No	Yes	No	No	Yes	No	No
TreeSet	Yes	Yes	No	No	No	No	No
LinkedHashSet	Yes	No	No	No	Yes	No	No
HashMap	No	Yes	Yes	No	Yes	No	No
TreeMap	Yes	Yes	Yes	No	No	No	No
LinkedHashMap	Yes	No	Yes	No	Yes	No	No
EnumSet	Yes	Yes	No	No	No	No	No
CopyOnWriteArrayList	Yes	Yes	No	Yes	Yes	Yes	No
CopyOnWriteArraySet	Yes	Yes	No	No	Yes	Yes	No
EnumMap	No	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	No	Yes
LinkedBlockingQueue	Yes	No	No	Yes	No	No	Yes
ArrayBlockingQueue	Yes	Yes	No	Yes	No	No	Yes/No
PriorityBlockingQueue	No	Yes	No	Yes	No	No	Yes
SynchronouseQueue	Yes	Yes	No	No	No	Yes	Yes

Задание 3. Ссылки на коллекции

Определена иерархия классов

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

Укажите корректные и некорректные операторы. Дайте ответу пояснение.

	correct	not correct
Doctor doctor1 = new Doctor();	v	
Doctor doctor2 = new MedicalStaff();		v
Doctor doctor3 = new HeadDoctor();	v	
Object object1 = new HeadDoctor();	v	
HeadDoctor doctor5 = new Object();		v

Doctor doctor6 = new Nurse();		v
Nurse nurse = new Doctor();		v
Object object2 = new Nurse();	v	
List<Doctor> list1= new ArrayList<Doctor>();	v	
List<MedicalStaff> list2 = new ArrayList<Doctor>();		v
List<Doctor> list3 = new ArrayList<MedicalStaff>();		v
List<Object> list4 = new ArrayList<Doctor>();		v
List<Object> list5 = new ArrayList<Object>();	v	

Задание 4. Применение коллекций

Заполните таблицу.

	Основная функциональность	Примеры типичного использования
Set	Хранит уникальные значения	HashSet, TreeSet, LinkedHashSet
List	Хранит списки на основе массива и двусвязного списка, в отличие от массива, может изменять свой размер.	ArrayList, LinkedList
Queue	Структура данных описывающая очередь, может быть реализована как FIFO и как LIFO	LinkedList, PriorityQueue, ArrayDeque
Map	Структура данных описывающая ключ – значение, ключи в картах должны быть уникальны.	HashMap, TreeMap, LinkedHashMap