

Практическое задание по теме

JAVA.SE.06 Generic and Collections

Выполнила
Бабеева Ольга

декабрь 2015

Задание 1. Классы коллекций

Изучите классы реализации коллекций и заполните таблицу:

		ordering	random access	key- value pairs	allows duplicates	allows null values	thread safe	blocking operations
Single-threaded collections								
List	ArrayList	by index	yes	no	yes	yes	no	no
	LinkedList ⁱ		no					
	Vector		yes				yes	
Set	HashSet	no	no	no	yes	no	no	
	LinkedHashSet	insertion						
	TreeSet	sorted						
	EnumSet	natural (declaration)			no			
	BitSet ⁱⁱ	by index						
Queues and dequeues								
	PriorityQueue	sorted	no	no	yes	no	no	
	ArrayDeque	FIFO						
	Stack	by index						yes
Map ⁱⁱⁱ	HashMap	no	no	yes	keys: no, values: yes	one null key, values: yes	no	no
	LinkedHashMap	insertion or last access						
	TreeMap	sorted				keys: no, values: yes		
	Hashtable	no				keys: no, values: no		
	EnumMap	natural (declaration)				keys: no, values: yes		
	IdentityHashMap	no				one null key, values: yes		
	WeakHashMap	no						
concurrent collections								
List	CopyOnWriteArrayList	by index	yes	no	yes	yes	yes	no
Set	ConcurrentSkipListSet	natural or comparator	no	no	no	no	yes	no
	CopyOnWriteArraySet	insertion				yes		
	Queue and deque							
	ArrayBlockingQueue	FIFO	no	no	yes	no	yes	yes
	ConcurrentLinkedDeque	LIFO						no
	ConcurrentLinkedQueue	FIFO						
	DelayQueue	no						
	LinkedBlockingDeque	LIFO						
	LinkedBlockingQueue	FIFO						
	LinkedTransferQueue	FIFO						
	PriorityBlockingQueue	sorted						
	SynchronousQueue	no						
Map	ConcurrentHashMap	no	no	yes	keys: no, values: yes	one null key, values: yes	yes	no

ⁱ *LinkedList* также реализует интерфейс *Deque*

ⁱⁱ BitSet не реализует интерфейс Collection

ⁱⁱⁱ Map не реализует интерфейс Collection, но входит в Java Collections Framework

Источники:

1. <https://docs.oracle.com/javase/7/docs/api/java/util/AbstractCollection.html>
2. <http://www.programcreek.com/2009/02/collection-interface-concrete-implementation-classes-summary-and-some-examples/>
3. https://en.wikiversity.org/wiki/Java_Collections_Overview

Задание 2. Использование Map

Ответьте на вопрос: как ведет себя map-коллекция если в нее добавить элемент с ключом, который уже присутствует?

В этом случае старое значение заменится на новое.

Источник: <http://docs.oracle.com/javase/8/docs/api/java/util/Map.html#put-K-V->

Задание 3. Ссылки на коллекции

Определена иерархия классов:

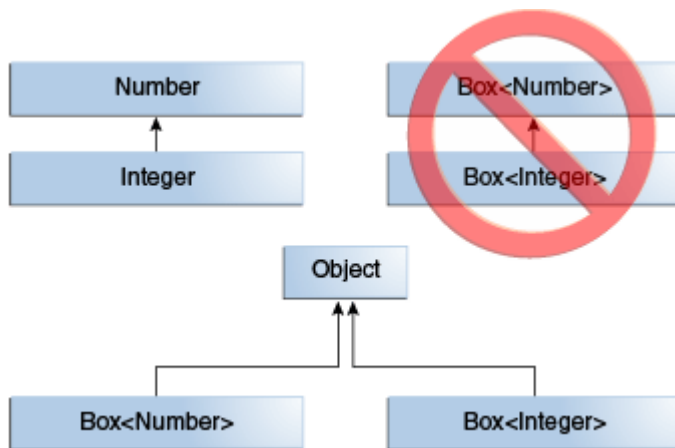
```
class MedicalStaff{}
class Doctor extends MedicalStaff{}
class Nurse extends MedicalStaff{}
class HeadDoctor extends Doctor{}
```

Укажите корректные и некорректные операторы. Дайте ответу пояснение.

	Оператор	Корректность
1	Doctor doctor1 = new Doctor();	верно
2	Doctor doctor2 = new MedicalStaff();	неверно
3	Doctor doctor3 = new HeadDoctor();	верно
4	Object object1 = new HeadDoctor();	верно
5	HeadDoctor doctor5 = new Object();	неверно
6	Doctor doctor6 = new Nurse();	неверно
7	Nurse nurse = new Doctor();	неверно
8	Object object2 = new Nurse();	верно
9	List<Doctor> list1= new ArrayList<Doctor>();	верно
10	List<MedicalStaff> list2 = new ArrayList<Doctor>();	неверно
11	List<Doctor> list3 = new ArrayList<MedicalStaff>();	неверно
12	List<Object> list4 = new ArrayList<Doctor>();	неверно
13	List<Object> list5 = new ArrayList<Object>();	верно

1. Просто создаём объект класса Doctor.
2. MedicalStaff – суперкласс для класса Doctor, объект класса MedicalStaff не является объектом класса Doctor.
3. HeadDoctor – подкласс класса Doctor, любой объект HeadDoctor является объектом класса Doctor.
4. Object – суперкласс всех классов, объект любого класса является объектом класса Object.

5. HeadDoctor – подкласс класса Object, объект класса Object не является объектом класса HeadDoctor.
6. Doctor и Nurse – подклассы MedicalStaff, объект класса Doctor не является объектом класса Nurse.
7. Doctor и Nurse – подклассы MedicalStaff, объект класса Nurse не является объектом класса Doctor.
8. Объект любого класса (в том числе и Nurse) является объектом класса Object.
9. Просто создаём List для объектов класса Doctor.
10. Объект класса ArrayList<Doctor> не является объектом класса List<MedicalStaff>. См. схему ниже.
11. Объект класса ArrayList<MedicalStaff> не является объектом класса List<Doctor>. См. схему ниже.
12. Объект класса ArrayList<Doctor> не является объектом класса List<Object>. См. схему ниже.
13. Просто создаём List для объектов класса Object.



Источник: <https://docs.oracle.com/javase/tutorial/java/generics/inheritance.html>

Задание 4. Применение коллекций

Заполните таблицу:

	Основная функциональность	Примеры типичного использования
Set	Коллекция, которая не может содержать повторяющиеся элементы. Моделирует математическое множество, используется для представления таких наборов, как например карты на руке у игрока в покер, предметы в расписании студента.	<u>Множество ключевых слов Java:</u> <pre>Set<String> keyWordsSet = Collections.unmodifiableSet(new LinkedHashSet<>() Arrays.asList(keyWordStrings));</pre>
List	Упорядоченная коллекция, может содержать повторяющиеся элементы. Пользователь Lista в общем случае имеет чёткий контроль над тем, в какую позицию вставляется новый элемент и может получать элементы по индексу (позиции).	<u>Список фильмов:</u> <pre>List<Movie> collection = new ArrayList<>();</pre>
Queue	Коллекция, которая используется для хранения множества элементов перед их обработкой. Предоставляет дополнительные операции вставки,	<u>Очередь для подсчёта площади пересекающихся прямоугольников с помощью сканирующей прямой:</u>

Map

извлечения и проверки элементов. Queue обычно, но не обязательно содержат элементы в порядке FIFO (первый пришёл – первый ушёл). Каждая реализация Queue должна указывать порядок, в котором хранятся элементы.

```
class Event implements Comparable<Event>
{ //... }

PriorityQueue<Event> queue
    = new PriorityQueue<>();
for (Rectangle rectangle : rectangles){
    queue.add(new Event(rectangle));
}
```

Объект, который сопоставляет ключам значения. Map не может содержать повторяющиеся ключи, каждый ключ соответствует не более, чем одному значению.

[Список оценок по предметам:](#)

```
Map<Course, List<Number>> courseList
    = new HashMap<>();
```