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
Списки List							
ArrayList	Yes	Yes	No	Yes	Yes	No	No
LinkedList	Yes	No	No	Yes	Yes	No	No
Vector	Yes	Yes	No	Yes	Yes	Yes	No
Stack	Yes	Yes	No	Yes	Yes	Yes	No
Множества Set							
HashSet	No	No	No	No	Yes	No	No
LinkedHashSet	Yes	No	No	No	Yes	No	No
TreeSet	Yes	No	No	No	No	No	No
Карты отображения Map							
LinkedHashMap	Yes	No	Yes	No	Yes	No	No
HashMap	No	No	Yes	No	Yes	No	No
TreeMap	Yes	No	Yes	No	Keys - No Values - Yes	No	No
Очереди Queue							
ArrayDeque	Yes	No	No	Yes	No	No	No
PriorityQueue	Yes	No	No	Yes	No	No	No

Задание 2. Использование Map
Ответьте на вопрос: как ведет себя map-коллекция если в нее добавить элемент с ключом, который уже присутствует?

Ответ: При добавлении элемента с ключом, который уже присутствует, присваивается новое значение ключа.

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(); Верно, так как мы можем создавать объект и ссылку совместимого типа.	✔	
Doctor doctor2 = new MedicalStaff(); Неверно, в данном случае объект и ссылка не совместимы, так как MedicalStaff находится выше в иерархии наследования		✔
Doctor doctor3 = new HeadDoctor(); Верно, так как Doctor находится в иерархии наследования выше, чем HeadDoctor.	✔	
Object object1 = new HeadDoctor(); Класс Object является классом, от которого наследуются все классы, в том числе HeadDoctor, поэтому конфликта не происходит.	✔	
HeadDoctor doctor5 = new Object(); Object находится в иерархии наследования выше, чем HeadDoctor, поэтому неверно		✔
Doctor doctor6 = new Nurse(); Ссылке можно присвоить только объект такого же типа или подтипа (класса наследника), но не того класса, который имеет общего предка. Все классы наследуются от Object, все классы имеют общий		✔

родительский класс. Нужна возможность создавать классы, которые существуют отдельно друг от друга. Если бы данный пример был бы верным, то это было бы следствием того, что абстракция не соблюдается.		
Nurse nurse = new Doctor(); Аналогично предыдущему примеру.		✔
Object object2 = new Nurse(); Верно, поскольку Object является родительским классом.	✔	
List<Doctor> list1= new ArrayList<Doctor>(); List находится выше в иерархии классов, чем ArraList, передаваемые параметры одного типа, конфликта не происходит.	✔	
List<MedicalStaff> list2 = new ArrayList<Doctor>(); Подобное хранение недопустимо, поскольку нельзя гарантировать, что в список будут добавлены объекты допустимого типа. Коллекции “не знают” тип хранимых элементов, эта информация “стирается” во время компиляции. Поэтому они не могут создавать исключение во время выполнения в случае хранения недопустимых типов. Вместо этого исключение ClassCastException будет выброшено в некоторой далекой, трудно ассоциируемой точке кода, когда значение будет считано из коллекции.		✔
List<Doctor> list3 = new ArrayList<MedicalStaff>(); Аналогично тому, что выше.		✔
List<Object> list4 = new ArrayList<Doctor>(); Аналогично тому, что выше.		✔
List<Object> list5 = new ArrayList<Object>(); List находится выше в иерархии классов, чем ArraList, передаваемые параметры одного типа, конфликта не происходит.	✔	

Задание 4. Применение коллекций

Заполните таблицу.

	Основная функциональность	Примеры типичного использования

Set	Специализирует коллекции для обработки множеств, содержащих уникальные элементы. Добавляет запрет на дублирующиеся элементы.
	
	int size();
	boolean isEmpty();
	boolean contains(Object element);
	boolean add(E element);
	boolean remove(Object element);
		Создание набора объектов. Например, для помещения ключей map (метод keySet()) с целью последующего перебора.

List	Специализирует коллекции для обработки списков. Сохраняет последовательность добавления элементов и позволяет осуществлять доступ к элементу по индексу.
	
	E get(int index);
	boolean add(E element);
	void add(int index, E element);
	E remove(int index);
		Организация хранения данных, к которым нужно получить доступ по индексу, например, набор канцелярских товаров, хранение значений пикселей.
		Его используют там, где нужно реализовать список.

Queue	Расширяет коллекцию методами для вставки, выборки и просмотра элементов. Кроме базовых методов Collection очередь(Queue) предоставляет дополнительные методы по добавлению, извлечению и проверке элементов. E element(); boolean offer(E o); E peek(); E poll(); E remove();	Применяется в тех ситуациях, в которых требуется размещение элемента перед его обработкой.
Map	Map работает с наборами пар объектов «ключ-значение» Все ключи в картах уникальны.	Реализация словаря. Хранение средних оценок студентов, например, как в задании 5 2-ого модуля.

	V put(K key, V value);
	V get(Object key);
	V remove(Object key);
	boolean containsKey(Object key);
	boolean containsValue(Object value);	