

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
General-Purpose Implementations							
HashSet	No	No	No	No	Yes	No	Yes
TreeSet	Yes	No	No	No	No	No	Yes
LinkedHashSet	No	No	No	No	Yes	No	Yes
ArrayList	Yes	Yes	No	Yes	Yes	No	Yes
LinkedList	Yes	Yes	No	Yes	Yes	No	Yes
HashMap	No	Yes	Yes	No	No	No	Yes
TreeMap	Yes	Yes	Yes	No	No	No	Yes
LinkedHashMap	No	Yes	Yes	No	No	No	Yes
PriorityQueue	No	No	No	Yes	No	No	Yes
Stack	No	No	No	Yes	No	No	Yes
Vector	No	Yes	No	Yes	Yes	Yes	Yes
HashTable	No	Yes	Yes	No	Yes	No	Yes
Special-Purpose Implementations							
EnumSet	No	No	No	No	Yes	No	Yes
CopyOnWriteArraySet	No	No	No	No	Yes	Yes	Yes
CopyOnWriteArrayList	Yes	Yes	No	Yes	Yes	Yes	Yes
EnumMap	No	Yes	Yes	No	Yes	No	Yes
WeakHashMap	No	Yes	Yes	No	No	No	Yes
IdentityHashMap	No	Yes	Yes	No	No	No	Yes
Concurrent implementations							
ConcurrentHashMap	No	Yes	Yes	No	No	No	No
LinkedBlockingQueue	No	No	No	Yes	No	No	No
ArrayBlockingQueue	No	No	No	Yes	No	No	No
PriorityBlockingQueue	No	No	No	Yes	No	No	No
DelayQueue	No	No	No	Yes	No	No	Yes
SynchronousQueue	No	No	No	Yes	No	No	Yes
LinkedTransferQueue	No	No	No	Yes	No	No	Yes

Задание 2. Использование Map

Сделано.

Legal Notice

This document contains privileged and/or confidential information and may not be disclosed, distributed or reproduced without the prior written permission of EPAM Systems.

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
// Создание экземпляра класса Doctor doctor1 = new Doctor();	✓	
// Конструктор MedicalStaff() возвращает объект класса MedicalStaff, который не может быть приведен к классу Doctor («ничего не знает о нем») Doctor doctor2 = new MedicalStaff();		✓
// Объект класса HeadDoctor может быть приведен к классу Doctor, т.к. класс HeadDoctor наследует Doctor Doctor doctor3 = new HeadDoctor();	✓	
// Верно, т.к. все классы - наследники Object Object object1 = new HeadDoctor();	✓	
// Нельзя привести к типу («класс Object ничего не знает о HeadDoctor») HeadDoctor doctor5 = new Object();		✓
// Классы имеют общего родителя, но между собой не связаны Doctor doctor6 = new Nurse();		✓
// Аналогично Nurse nurse = new Doctor();		✓
// Верно, т.к. все классы - наследники Object Object object2 = new Nurse();	✓	

	correct	not correct
// Верно List<Doctor> list1= new ArrayList<Doctor>();	✓	
// Нельзя привести к типу List<MedicalStaff> list2 = new ArrayList<Doctor>();		✓
List<Doctor> list3 = new ArrayList<MedicalStaff>();		✓
// Нельзя привести к типу List<Object> list4 = new ArrayList<Doctor>();		✓
List<Object> list5 = new ArrayList<Object>();	✓	

Задание 4. Применение коллекций

Заполните таблицу.

	Основная функциональность	Примеры типичного использования
Set	Обработка множеств, содержащих уникальные элементы	Обработка набора уникальных объектов (список сотрудников компании, перечень небесных тел солнечной системы)
List	Обработка списков: - могут включать одинаковые элементы; - сохранение порядка элементов; - доступ к любому элементу.	Список дел Список покупок
Queue	Обработка очереди по принципу FIFO	Обработка сообщений, логов, новостей (новые сообщения помещаются в конец очереди, для обработки выбираются сообщения из начала)
Map	Обработка коллекций, хранимых в виде пар ключ-значение	Словарь Данные из файла properties

Legal Notice

This document contains privileged and/or confidential information and may not be disclosed, distributed or reproduced without the prior written permission of EPAM Systems.