**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 |  |
| LinkedList | Yes | Yes | No | Yes | Yes | No |  |
| HashSet | No | No | No | No | Yes | No |  |
| LinkedHashSet | Yes | No | No | No | Yes | No |  |
| TreeSet | Yes | No | No | No | No | No |  |
| HashMap | No | By key | Yes | Only values | Yes | No |  |
| LinkedHashMap | Yes | By key | Yes | Only values | Yes | No |  |
| TreeMap | Yes | By key | Yes | Only values | No | No |  |

**Задание 2. Использование Map**

  Создать “универсальный” класс, позволяющий получить значение из любого properties-файла. Физическое чтение файла должно происходить только один раз. Результаты чтения храните в коллекции типа Map. Ответьте на вопрос: как ведет себя map-коллекция если в нее добавить элемент с ключом, который уже присутствует?

**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(); | + |  |
| Doctor doctor2 = **new** MedicalStaff(); |  |  |
| Doctor doctor3 = **new** HeadDoctor(); |  | + |
| Object object1 = **new** HeadDoctor(); | + |  |
| HeadDoctor doctor5 = **new** Object(); |  |  |
| Doctor doctor6  = **new** Nurse(); |  |  |
| Nurse nurse = **new** Doctor(); |  |  |
| Object object2 = **new** Nurse(); | + |  |
|  |  |  |
| List<Doctor> list1= **new** ArrayList<Doctor>(); | + |  |
| List<MedicalStaff> list2 = **new** ArrayList<Doctor>(); |  | + |
| List<Doctor> list3 = **new** ArrayList<MedicalStaff>(); |  | + |
| List<Object> list4 = **new** ArrayList<Doctor>(); |  | + |
| List<Object> list5 = **new** ArrayList<Object>(); | + |  |

**Задание 4. Применение коллекций**

 Заполните таблицу.

|  |  |  |
| --- | --- | --- |
|  | Основная функциональность | Примеры типичного использования |
| Set | Эл-ты не повторяются. Нет произвольного доступа к эл. | Предназначены для хранения множества неповторяющихся объектов. |
| List | Эл-ты могут повторяться. Возможна вставка и удаление эл. в любом месте списка. Доступ к эл. по индексу. | Работа с произвольным списком объектов, которые могут повторяться. |
| Queue | (FIFO)Добавление в начало. Удаление с конца. | Организация связи  Producers  |  Queue  |  Сustomers |
| Deque | Линейный набор, который поддерживает вставку элемента и удаление в обоих концах. |
| Map | Map<K,V> хранит пары "ключ-значение". Каждое значение можно найти по его ключу. | В Map могут находиться записи с информацией о сотрудниках, где ключами являются идентификационные номера сотрудников. |