|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Ordering | Random Access | Key-Value Pairs | Allows Duplicates | Allows Null Values | Thread Safe | Blocking Operations |
| ArrayList | y | y | n | y | y | n | n |
| LinkedList | y | y | n | y | y | n | n |
| TreeMap | y sort | n | y | y (value) | y | n | n |
| HashMap | n | n | y | y (value) | y | n | n |
| LinkedHashMap | y | n | y | y (value) | y | n | n |
| HashSet | n | n | n | n | y | n | n |
| TreeSet | y sort | n | n | n | y(1) | n | n |
| LinkedHashSet | y | n | n | n | y | n | n |
| PriorityQueue | y sort | n | n | y | n | n | n |
| ArrayDeque | y sort | n | n | y | n | n | n |

6.1

6.3

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 |  |

*Нельзя создавать экземпляр предка. При объявлении листа тип слева и справа должен совпадать точно.*

6.4

|  |  |  |
| --- | --- | --- |
|  | Основная функциональность | Примеры типичного использования |
| Set | Список не повторяющихся значений | Список моделей, имен (как типов), др. сущ. не повторяющихся в наборе |
| List | Список значений с доступом по индексу | Хранение данных о пользователях, товарах и др. бизнес-сущностях |
| Queue | СД FIFO/LIFO | Кэширование входящих данных |
| Map | Хранилище пар ключ-значение | Хранение списков пользователей вида id-username и др. сущ. идентифицируемых одним параметром |