|  |
| --- |
| EPAM Systems, RD Dep., RD Dep. |
| Практическое задание  SFT.01.001 Планирование и организация процесса тестированияGeneric and Collections |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| REVISION HISTORY | | | | | |
| Ver. | Description of Change | Author | Date | Approved | |
| Name | Effective Date |
| <1.0> | Первая версия | Игорь Блинов | <04.08.2011> |  |  |
| <2.0> | Вторая версия: задания изменены согласно обновленному содержанию модуля | Ольга Смолякова | <12.11.2014> |  |  |

##### **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 | no | yes | no | yes | yes | no | no |
| LinckedList | no | no | no | yes | yes | no | no |
| TreeMap | yes | no | yes | no | no | no | no |
| HashTable | yes | no | yes | no | no | yes | yes |
| HashMap | yes | no | yes | no | yes | no | no |
| HashSet | yes | no | no | no | yes | no | no |
| TreeSet | yes | no | no | no | no | no | no |
| ArrayDeque | no | no | no | yes | yes | no | no |
| LinckedHashSet | no | no | no | no | yes | no | no |
| PriorityQueue | yes | no | no | yes | no | 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?

Ответ: Новый элемент будет перезаписан вместо старого, который можно было получить по этому ключу.

##### **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 | Множество ─ коллекция без повторяющихся  элементов  Интерфейс Set<E> содержит методы, унаследованные  Collection<E> и добавляет запрет на дублирующиеся  элементы. | Операции со множествами. Хранилища для уникальных значений |
| List | упорядоченная коллекция (иногда называется sequence)  Список может содержать повторяющиеся элементы.  Интерфейс List сохраняет последовательность добавления элементов и позволяет  осуществлять доступ к элементу по индексу. | Расширяемые массивы данных |
| Queue | предназначенная для размещения элемента перед его обработкой. Расширяет коллекцию методами для вставки, выборки и просмотра элементов  Очередь – хранилище элементов, предназначенных для обработки.  Кроме базовых методов Collection очередь(Queue) предоставляет дополнительные  методы по добавлению, извлечению и проверке элементов.  Чаще всего порядок выдачи элементов соответствует FIFO (first-in, first-out), но в  общем случае определяется конкретной реализацией.  Очереди не могут хранить null.  У очереди может быть ограничен размер. | Парсинг формул, тегов HTML страниц |
| Map | Интерфейс Map работает с наборами пар объектов «ключ-значение»  Все ключи в картах уникальны.  Уникальность ключей определяет реализация метода equals(...).  Для корректной работы с картами необходимо переопределить методы equals(...) и  hashCode(), допускается добавление объектов без переопределения этих методов, но найти  эти объекты в Map вы не сможете. | Properties |