

Домашнее задание 5.

Срок сдачи: 23:59 29 мая, 2015

1 Условие

Напишите многопоточный сервис, который позволял бы осуществлять сериализацию/десериализацию специальных классов. Эти классы

- имеют открытый конструктор без параметров
- обладают набором «свойств»
- некоторые «свойства» рассчитаны только на чтение, некоторые – на чтение и запись
- значение «свойства» может быть получено через соответствующий `get` метод без аргументов (`String getUsername()`)
- значение каждого «свойства», доступного для записи, может быть задано через соответствующий `set` метод с одним аргументом (`void setUsername(String userName)`)
- все «свойства» имеют примитивный тип или являются строками
- «свойства» могут быть никак не связаны с полями в классе! Например, объекты класса `Student` могут хранить значения «свойства» `surname` в поле `myFamilyName`.
- помните, что помимо `getters & setters` в классе могут быть другие `public` методы

Класс сервиса – `DistributedSerializator<T>`:

- обладает неблокирующим методом `Future<Boolean> serialize(T o, String name)`
Метод производит «сериализацию» в файл `name.properties`
Возвращаемое значение содержит `True` или `False` в зависимости от того оказалась ли сериализация успешной.
Если сериализация была прервана (`interrupt` или `cancel`) файл не должен быть создан.
- **Если в процессе сериализации произошло исключение, то файл тоже не должен быть создан. (Смысл этого пункта в том, чтобы в процессе работы не возникало файлов, которые хранят «невалидное» состояние)**
- обладает неблокирующим методом `Future<T> deserialize(String name)`
Метод производит «десериализацию» из файла `name.properties`
- обладает методом `T get(String name)`, позволяющим получить объект, десериализованный из файла `name.properties`. Если объект еще не был десериализован, то сначала он должен быть загружен из файла. Если уже был, то должна быть возвращена ссылка на уже загруженный объект.
Метод `get` должен быть написан таким образом, чтобы его могли вызывать несколько потоков одновременно **в не блокирующем режиме, если объект уже десериализован.**
- для проведения сериализации и десериализации суммарно выделено 10 потоков

При сериализации имена «свойств» класса сопоставляются с названиями полей в `.properties` файле путем удаления приставок `set-` или `get-`.

При сериализации необходимо сохранить все свойства объекта.

При десериализации восстанавливаются только свойства, доступные на запись (имеющие `set`

метод).

Обратите внимание, что, если в момент сериализации/десериализации (т.е. в процессе работы с файлом), другой поток пытается изменить/прочитать этот же файл (так как выполняет сериализацию/десериализацию), то он должен дожидаться освобождения файла и выполнить свою задачу (а не прерваться с исключением).

Для работы с файлами .properties используйте стандартный класс Properties.

При сериализации в файл необходимо сохранять информацию о том какой класс сериализуется. Это делается для того, чтобы при сериализации можно было знать объект какого класса создавать и сравнивать с тем, что мы хотим получить (напомню, что класс DistributedSerializator параметризован типом, который хочется получить).