

Домашнее задание 6. (De)Serialization via Reflection.

Срок сдачи: конец семестра

Прежде чем приступить к написанию, прочтите раздел **Замечания**.

1 Условие

Теперь когда вы знаете что такое reflection и с чем его едят, попробуем применить его для сериализации/десериализации объектов.

Классы с которыми мы будем работать подчиняются следующим условиям:

- имеют public конструктор без аргументов
- обладают заранее неизвестным набором “свойств”
- некоторые из свойств рассчитаны только на чтение, некоторые и на чтение и на запись
- значение каждого “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” объекта имеют примитивные типы (или **обертки** над ними) или являются строками.

Необходимо реализовать сериализацию “свойств” таких классов в файлы формата .properties и их десериализацию.

Имена “свойств” класса сопоставляются с названием полей в .properties файле следующим нехитрым способом: get/set удаляется, первая буква оставшейся части приводится к нижнему регистру.

Для простоты, можно сохранять в файл значения всех свойств класса. При загрузке, естественно, принимаются во внимание лишь свойства, доступные на запись (имеющие set метод).

Ваша задача:

- создать класс **ReflectionSerializer**, имеющий метод **void serialize(T t, String filename)**, осуществляющий сериализацию объекта типа **T** в файл
- создать класс **ReflectionDeSerializer**, имеющий метод **void deserialize(String filename, Class<T> clazz)**, осуществляющий десериализацию объекта типа **T** из файла
- создать класс **Student**, содержащий следующие “свойства” (не путать с полями), доступные и на чтение и на запись:

- **String name**
 - **String surname**
 - **int age**
 - **double avgGrade**
- **Main** должен “жульничать” со средним баллом: аргументом передается `.properties` файл, программа десериализует этого студента, увеличивает его средний балл на 1.0 (но не выше 5.0) и сериализует обратно в тот же файл.

2 Замечания

1. Классы, для которых выполняются перечисленные условия (за исключением последнего), а также условие сериализуемости и чуть более сложные конвенции именования методов, называют JavaBeans.
2. Для работы с `.properties` файлами используйте класс **Properties** из стандартной библиотеки. Будьте аккуратны, он является известнейшим примером плохо спроектированного класса! =)
3. Имена полей сериализуемых классов могут не соответствовать именам “свойств” класса (`get/set` методам). Например, объекты класса `Student` могут хранить значение “свойства” `surname` в поле `myFamilyName`.
4. На всякий случай, `get`-еры и `set`-еры обязательно начинаются на “`get`” и “`set`” соответственно, а следующая буква заглавная =)
5. Помимо `get` и `set`, в классе могут быть и другие `public` методы.
6. Реализуемая здесь функциональность не имеет ничего общего со стандартными механизмами сериализации в языке Java (в частности, интерфейс **Serializable** не имеет к происходящему никакого отношения =)).
7. К сожалению, в Java нет “синтаксического сахара” для работы со свойствами