**Java I\O**

1. Что такое сериализация, для чего нужна, когда применяется? Ключевое слово transient, для чего нужно? Сериализация static-полей.

**Сериализация (Serialization)** — это процесс, который переводит объект в последовательность байтов, по которой затем его можно полностью восстановить. Зачем это нужно? Дело в том, при обычном выполнении программы максимальный срок жизни любого объекта известен — от запуска программы до ее окончания. Сериализация позволяет расширить эти рамки и «дать жизнь» объекту так же между запусками программы.

Дополнительным бонусом ко всему является сохранение кроссплатформенности. Не важно какая у вас операционная система, сериализация переводит объект в поток байтов, который может быть восстановлен на любой ОС.

Реализовать механизм сериализации довольно просто. Необходимо, чтобы ваш класс реализовывал интерфейс **Serializable (интерфейс-маркер)**. Это интерфейс — идентификатор, который не имеет методов, но он указывает jvm, что объекты этого класса могут быть сериализованы. Так как механизм сериализации связан с базовой системой ввода/вывода и переводит объект в поток байтов, для его выполнения необходимо создать выходной поток OutputStream, упаковать его в ObjectOutputStream и вызвать метод writeObject(). Для восстановления объекта нужно упаковать InputStream в ObjectInputStream и вызвать метод readObject().

Некоторые объекты не хранят все свои данные в себе, а лишь ссылаются на другие объекты и/или источники данных. Например, консоль (System.in), поток ввода (InputStream), или что-нибудь еще. Поэтому их нельзя сериализировать.

Пример:



(восстановление из файла – десериализация)

В Java есть ключевое слово **transient**. Оно применяется к полям класса и означает, что данное поле не входит в персистентное состояние класса. Т.е. при сериализации данное поле не будет записываться. И, соответственно, при де-сериализации оно не будет восстанавливаться из потока байтов.  
  
Применение: **private transient <member-variable>;**  
или  
**transient private <member-variable>;**  
  
Пример кода:

**public** **class** Foo {  
 **private** **String** saveMe;  
 **private** **transient** **String** dontSaveMe;  
 **private** **transient** **String** password;  
 *//...*  
}

При стандартной сериализации поля, имеющие модификатор *static*, не сериализуются. Соответственно, после десериализации это поле значения не меняет. При использовании реализации ***Externalizable*** (расширяет Serializable и добавляет методы записи и чтения writeExternal и readExternal) сериализовать и десериализовать статическое поле можно, но не рекомендуется этого делать, т.к. это может сопровождаться трудноуловимыми ошибками.

1. (сериализация)Возможно ли сохранить объект не в байт-код, а в xml-файл?
2. Что такое ClassLoader? Если изменить static переменную в классе, загруженном одним ClassLoader, что будет видно в том же классе, загруженном другим. Возможно ли синглтоны создавать несколько раз?