

Тестовое задание №3 - custom serializer

- JsonSerializer - это maven-проект, содержащий реализацию сериализатора и тесты.
- JsonSerializerExample - приложение для тестирования сериализатора.

Пример использования сериализатора:

```
OutputStream out = null;
InputStream in = null;
try {
    //reading input file
    Serializer persister = new Persister();
    Ship ship = persister.read(Ship.class, new File(args[0]));
    //serializing to binary data
    SerializerCollection collection = SerializerCollection.getDefault();
    out = new DataOutputStream(new BufferedOutputStream(new FileOutputStream(args[1] + ".bin")));
    collection.toStream(collection, ship, out);
    out.close();
    //deserializing object
    in = new DataInputStream(new BufferedInputStream(new FileInputStream(args[1] + ".bin")));
    Ship ship2 = (Ship) collection.fromStream(collection, in, Ship.class, null);
    in.close();
} catch (Exception ex) {
} finally {
    if (out != null) {
        try {
            out.close();
        } catch (Exception ex) {
        }
    }
    if (in != null) {
        try {
            in.close();
        } catch (Exception ex) {
        }
    }
}
```

Для запуска теста сериализации служат файлы start.sh\start.bat проекта JsonSerializerExample.

Я добавил в manifest classpath и главный класс, так что можно и просто выполнить:

java -jar JsonSerializerExample-1.0-SNAPSHOT.jar входной_файл выходной_файл

JsonSerializerExample содержит пример сериализуемого графа в файле input.xml

```
<ship>
  <pilots class="java.util.LinkedList">
    <pilot>
      <id>01</id>
      <name>Arboc Digambara</name>
      <skillpoints>1123</skillpoints>
    </pilot>
    <pilot>
      <name>Maya Lightbringer</name>
      <skillpoints>1156</skillpoints>
    </pilot>
  </pilots>
  <modules class="java.util.HashMap">
    <entry>
      <string>cannon</string>
      <module>
        <name>1400 Artillery II</name>
        <description>Minmatar rulez!</description>
      </module>
    </entry>
    <entry>
      <string>forgot to install</string>
      <module/>
    </entry>
    <entry>
      <string>lazer</string>
      <module>
        <name>Super Mega Cannon Lazer</name>
        <description>Галактега опасности!11</description>
      </module>
    </entry>
  </modules>
</price>1020</price>
```

```
</ship>
```

По умолчанию выходной файл имеет имя output.bin.

Он содержит результат сериализации графа.

При тестовом запуске происходит сериализация графа, десериализация и сравнение, если все прошло успешно, то вывод будет примерно такой:

```
[root@arbocdi-nb JsonSerializerExample]# ./start.sh
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/data/projects/JavaSerializerExample/target/dependencies/logback-classic-1.1.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/data/projects/JavaSerializerExample/target/dependency/compile/logback-classic-1.1.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [ch.qos.logback.classic.util.ContextSelectorStaticBinder]
2016-07-31_20:19:53.337 [main] INFO net.sf.arbocdi.jsexample.Launcher:102 - Logger initialized
2016-07-31_20:19:53.458 [main] INFO net.sf.arbocdi.jsexample.Launcher:43 - Object from xml
  Ship(pilots=[Pilot(super=User(id=01), name=Arboc Digambara, skillpoints=1123), Pilot(super=User(id=null), name=Maya Lightbringer, skillpoints=1156)], modules={cannon=Module(name=1400 Artillery II, description=Minmatar rulez!), forgot to install=null, lazer=Module(name=Super Mega Cannon Lazer, description=Голактега опасносте!11)}, price=1020)
2016-07-31_20:19:53.465 [main] INFO net.sf.arbocdi.jsexample.Launcher:53 - Object from binary
  Ship(pilots=[Pilot(super=User(id=01), name=Arboc Digambara, skillpoints=1123), Pilot(super=User(id=null), name=Maya Lightbringer, skillpoints=1156)], modules={cannon=Module(name=1400 Artillery II, description=Minmatar rulez!), forgot to install=null, lazer=Module(name=Super Mega Cannon Lazer, description=Голактега опасносте!11)}, price=1020)
2016-07-31_20:19:53.466 [main] INFO net.sf.arbocdi.jsexample.Launcher:55 - Initial and serialized objects match true
2016-07-31_20:19:53.467 [main] INFO net.sf.arbocdi.jsexample.Launcher:60 - Closing streams
2016-07-31_20:19:53.467 [main] INFO net.sf.arbocdi.jsexample.Launcher:73 - Stopping logger

[root@arbocdi-nb JsonSerializerExample]#
```

Ограничения сериализатора

- Сериализатор не умеет работать с циклическими ссылками.
- Сериализатор не пишет в поток имя сериализуемого класса и соответственно при десериализации полагается только на информацию, содержащуюся в десериализуемых java классах.