Algorithms

Produced by

Eamonn de Leastar (edeleastar@wit.ie)

Department of Computing, Maths & Physics
Waterford Institute of Technology

http://www.wit.ie

http://elearning.wit.ie

Models

```
PacemakerAPI pacemakerAPI = new PacemakerAPI();

pacemakerAPI.createUser("Bart", "Simpson",   "bart@simpson.com", "secret");
pacemakerAPI.createUser("Homer", "Simpson",  "homer@simpson.com", "secret");
pacemakerAPI.createUser("Lisa", "Simpson", " lisa@simpson.com", "secret");

Collection<User> users = pacemakerAPI.getUsers();
System.out.println(users);

User homer = pacemakerAPI.getUserByEmail("homer@simpson.com");
System.out.println(homer);

pacemakerAPI.deleteUser(homer.id);
users = pacemakerAPI.getUsers();
System.out.println(users);
```

```
XStream xstream = new XStream(new DomDriver());
ObjectOutputStream out = xstream.createObjectOutputStream(new FileWriter("datastore.xml"));
out.writeObject(users);
out.close();
```

```java
XStream xstream = new XStream(new DomDriver());
ObjectOutputStream out = xstream.createObjectOutputStream(new FileWriter("datastore.xml"));
out.writeObject(users);
out.close();
```

- Outputs a collection of User Objects to an XML File called ‘datastore.xml’

```java
public class User
{
  static Long   counter = 0l;

  public Long   id;
  public String firstName;
  public String lastName;
  public String email;
  public String password;

  public Map<Long, Activity> activities = new HashMap<>();

  //...
}
```

```xml
<object-stream>
  <java.util.HashMap_-Values>
    <outer-class>
      <entry>
        <long>0</long>
        <models.User>
          <id>0</id>
          <firstName>Bart</firstName>
          <lastName>Simpson</lastName>
          <email>bart@simpson.com</email>
          <password>secret</password>
          <activities/>
        </models.User>
      </entry>
      <entry>
        <long>2</long>
        <models.User>
          <id>2</id>
          <firstName>Lisa</firstName>
          <lastName>Simpson</lastName>
          <email> lisa@simpson.com</email>
          <password>secret</password>
          <activities/>
        </models.User>
      </entry>
    </outer-class>
  </java.util.HashMap_-Values>
</object-stream>
```

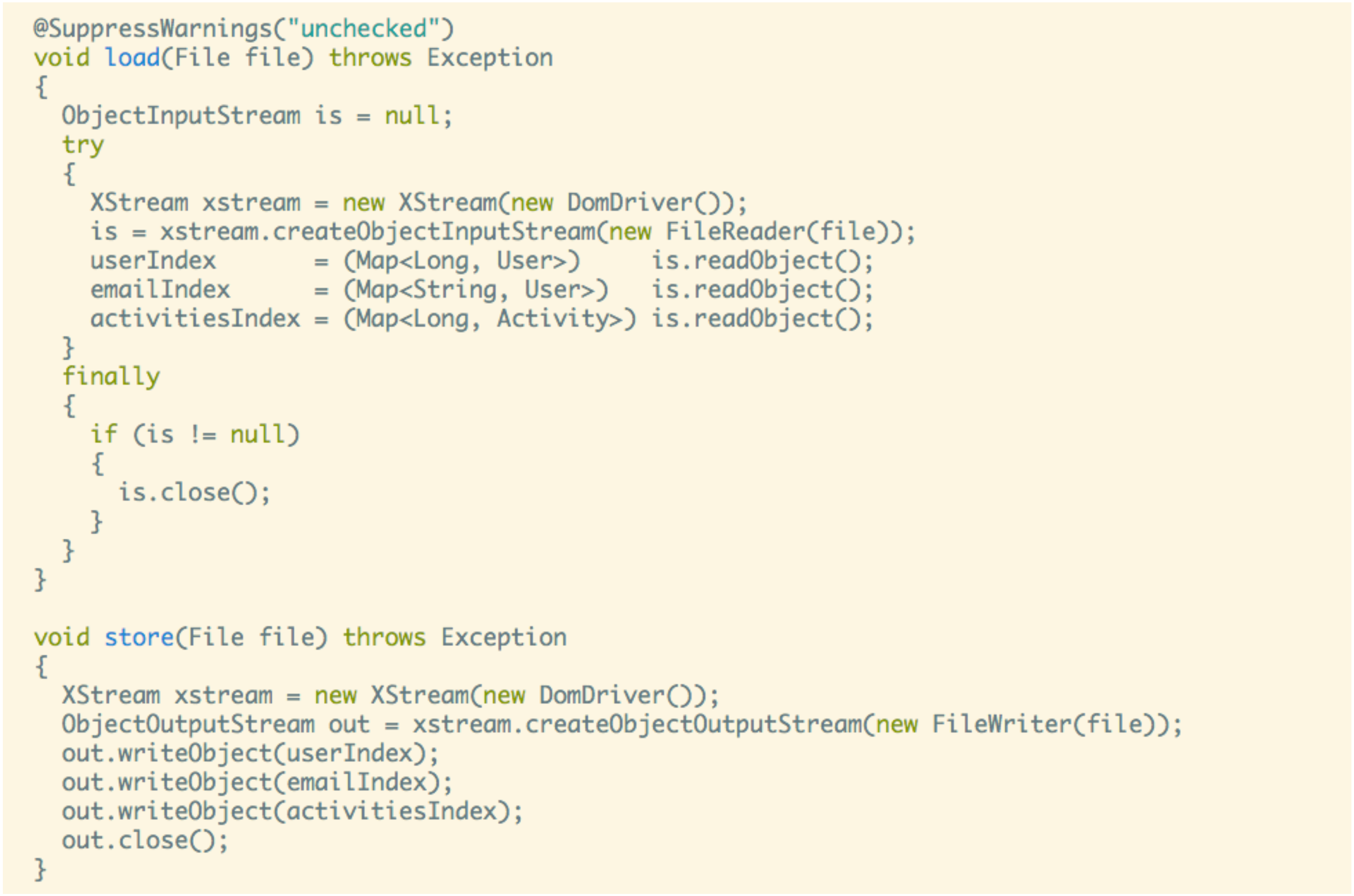
Pacemaker

- Three Collections:
  - Map of user ID -> User
  - Map of Email -> User
  - Make of Activity ID -> User

```
public class PacemakerAPI
{
  private Map<Long,   User>   userIndex       = new HashMap<>();
  private Map<String, User>   emailIndex      = new HashMap<>();
  private Map<Long, Activity> activitiesIndex = new HashMap<>();

  //...

  public Collection<User> getUsers ()
  {
    return userIndex.values();
  }

  public  void deleteUsers()
  {
    userIndex.clear();
    emailIndex.clear();
  }

  public void deleteUser(Long id)
  {
    User user = userIndex.remove(id);
    emailIndex.remove(user.email);
  }

  public Activity createActivity(Long id,         String type,
                                  String location, double distance)
  {
    Activity activity = null;
    Optional<User> user = Optional.fromNullable(userIndex.get(id));
    if (user.isPresent())
    {
      activity = new Activity (type, location, distance);
      user.get().activities.put(activity.id, activity);
      activitiesIndex.put(activity.id, activity);
    }
    return activity;
  }
```

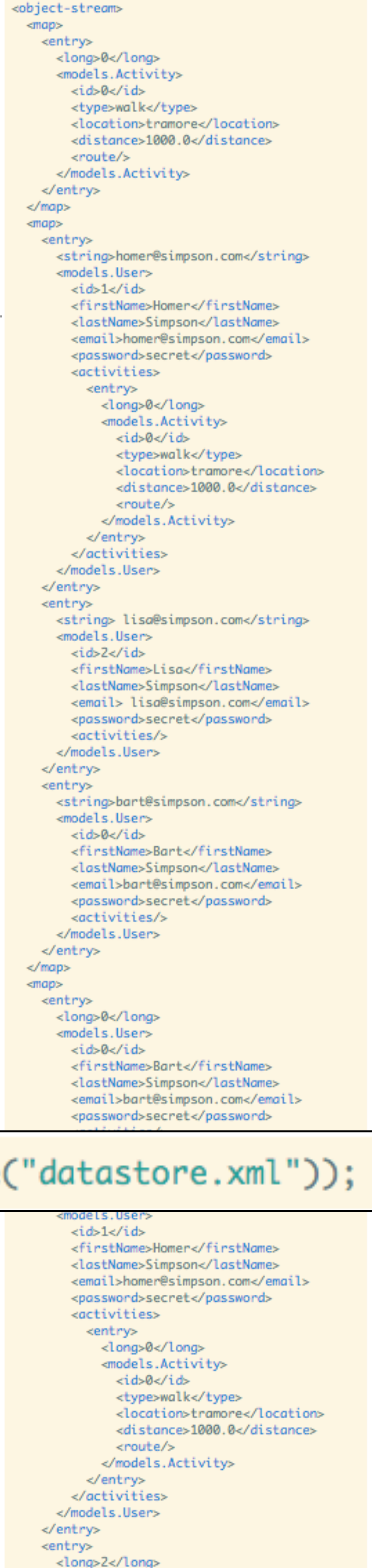
PacemakerAPI

- To introduce persistence capability into the api, we need two new methods”



```java
@SuppressWarnings("unchecked")
void load(File file) throws Exception
{
  ObjectInputStream is = null;
  try
  {
    XStream xstream = new XStream(new DomDriver());
    is = xstream.createObjectInputStream(new FileReader(file));
    userIndex       = (Map<Long, User>)     is.readObject();
    emailIndex      = (Map<String, User>)   is.readObject();
    activitiesIndex = (Map<Long, Activity>) is.readObject();
  }
  finally
  {
    if (is != null)
    {
      is.close();
    }
  }
}

void store(File file) throws Exception
{
  XStream xstream = new XStream(new DomDriver());
  ObjectOutputStream out = xstream.createObjectOutputStream(new FileWriter(file));
  out.writeObject(userIndex);
  out.writeObject(emailIndex);
  out.writeObject(activitiesIndex);
  out.close();
}
```

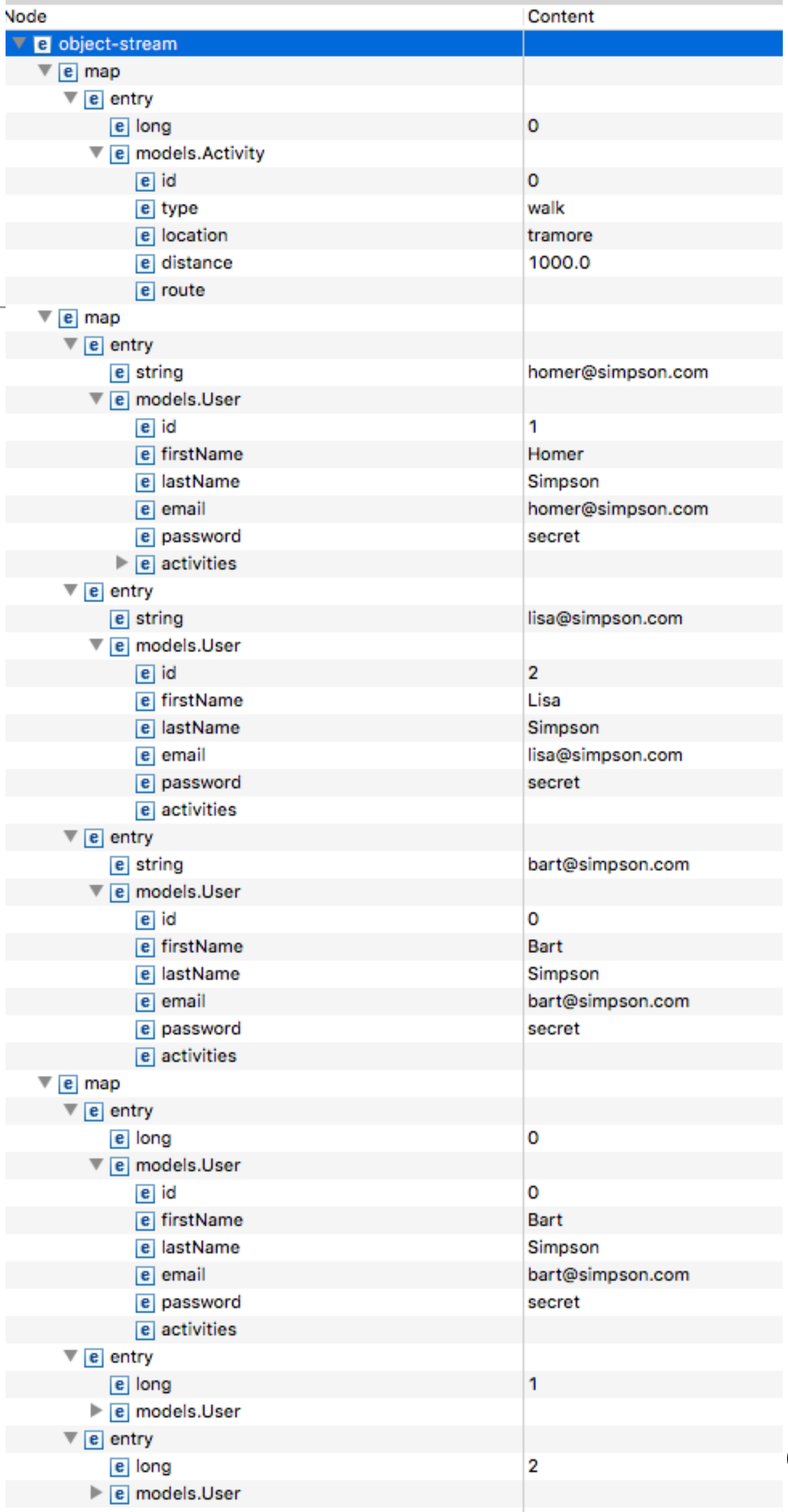
Sample



```xml
<object-stream>
  <map>
    <entry>
      <long>0</long>
      <models.Activity>
        <id>0</id>
        <type>walk</type>
        <location>tramore</location>
        <distance>1000.0</distance>
        <route/>
      </models.Activity>
    </entry>
  </map>
  <map>
    <entry>
      <string>homer@simpson.com</string>
      <models.User>
        <id>1</id>
        <firstName>Homer</firstName>
        <lastName>Simpson</lastName>
        <email>homer@simpson.com</email>
        <password>secret</password>
        <activities>
          <entry>
            <long>0</long>
            <models.Activity>
              <id>0</id>
              <type>walk</type>
              <location>tramore</location>
              <distance>1000.0</distance>
              <route/>
            </models.Activity>
          </entry>
        </activities>
      </models.User>
    </entry>
    <entry>
      <string> lisa@simpson.com</string>
      <models.User>
        <id>2</id>
        <firstName>Lisa</firstName>
        <lastName>Simpson</lastName>
        <email> lisa@simpson.com</email>
        <password>secret</password>
        <activities/>
      </models.User>
    </entry>
    <entry>
      <string>bart@simpson.com</string>
      <models.User>
        <id>0</id>
        <firstName>Bart</firstName>
        <lastName>Simpson</lastName>
        <email>bart@simpson.com</email>
        <password>secret</password>
        <activities/>
      </models.User>
    </entry>
  </map>
  <map>
    <entry>
      <long>0</long>
      <models.User>
        <id>0</id>
        <firstName>Bart</firstName>
        <lastName>Simpson</lastName>
        <email>bart@simpson.com</email>
        <password>secret</password>
```

```java
pacemakerAPI.store(new File("datastore.xml"));
```

```xml
      <models.User>
        <id>1</id>
        <firstName>Homer</firstName>
        <lastName>Simpson</lastName>
        <email>homer@simpson.com</email>
        <password>secret</password>
        <activities>
          <entry>
            <long>0</long>
            <models.Activity>
              <id>0</id>
              <type>walk</type>
              <location>tramore</location>
              <distance>1000.0</distance>
              <route/>
            </models.Activity>
          </entry>
        </activities>
      </models.User>
    </entry>
    <entry>
      <long>2</long>
```



```
Node                              Content
▼ e object-stream
  ▼ e map
    ▼ e entry
        e long                    0
      ▼ e models.Activity
          e id                    0
          e type                  walk
          e location              tramore
          e distance              1000.0
          e route
  ▼ e map
    ▼ e entry
        e string                  homer@simpson.com
      ▼ e models.User
          e id                    1
          e firstName             Homer
          e lastName              Simpson
          e email                 homer@simpson.com
          e password              secret
        ▶ e activities
    ▼ e entry
        e string                  lisa@simpson.com
      ▼ e models.User
          e id                    2
          e firstName             Lisa
          e lastName              Simpson
          e email                 lisa@simpson.com
          e password              secret
          e activities
    ▼ e entry
        e string                  bart@simpson.com
      ▼ e models.User
          e id                    0
          e firstName             Bart
          e lastName              Simpson
          e email                 bart@simpson.com
          e password              secret
          e activities
  ▼ e map
    ▼ e entry
        e long                    0
      ▼ e models.User
          e id                    0
          e firstName             Bart
          e lastName              Simpson
          e email                 bart@simpson.com
          e password              secret
          e activities
    ▼ e entry
        e long                    1
      ▶ e models.User
    ▼ e entry
        e long                    2
      ▶ e models.User
```

Generalising the Serializer

- An interface to encapsulate a general purpose serialiser

```
package utils;

public interface Serializer
{
  void push(Object o);
  Object pop();
  void write() throws Exception;
  void read() throws Exception;
}
```

XML Serializer - stack of objects to be read/written

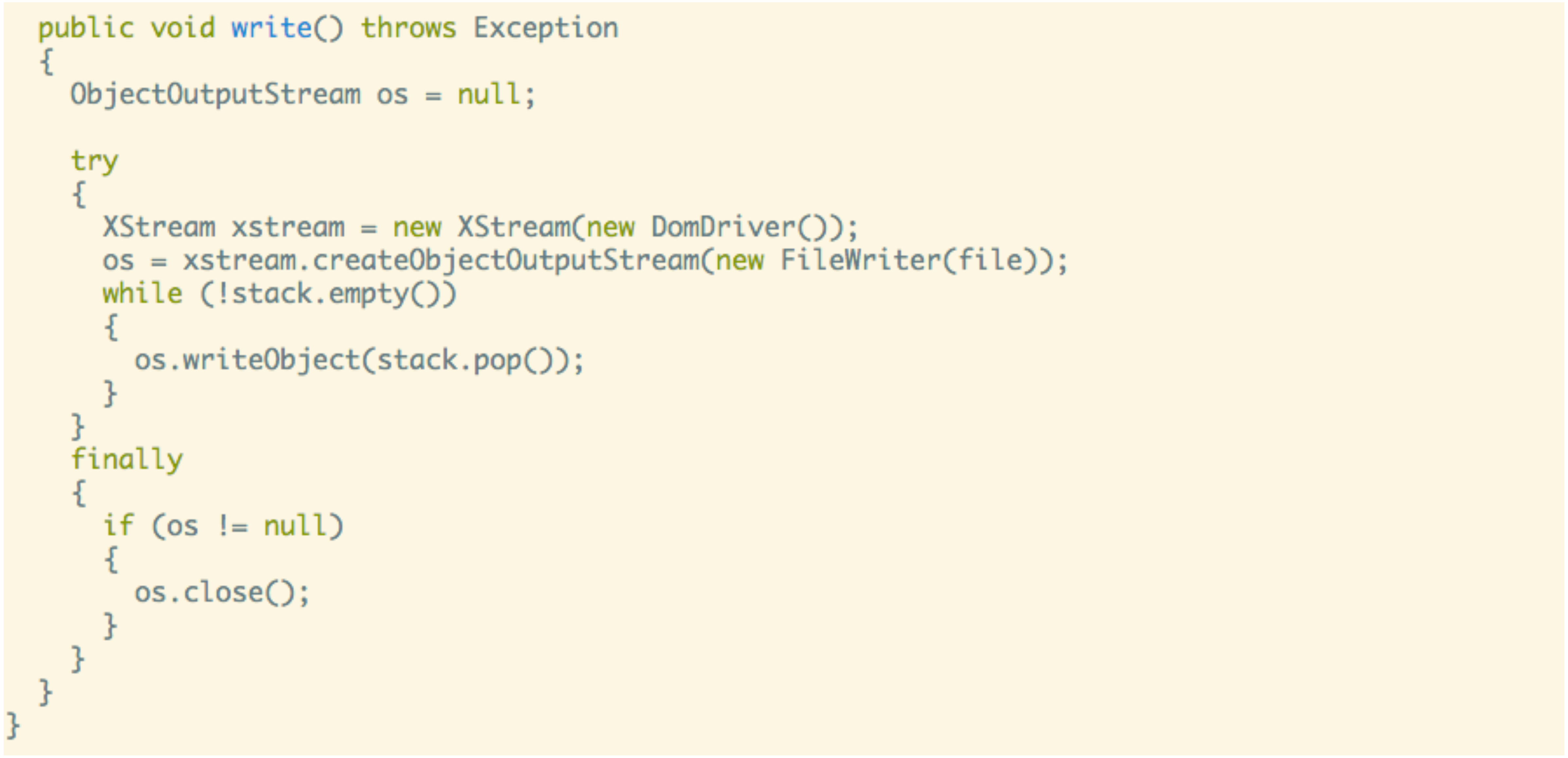
```
public class XMLSerializer implements Serializer
{

  private Stack stack = new Stack();
  private File file;

  public XMLSerializer(File file)
  {
    this.file = file;
  }

  public void push(Object o)
  {
    stack.push(o);
  }

  public Object pop()
  {
    return stack.pop();
  }
```

- push objects to be serialised onto a stack prior to write
- if read has taken place, pop read objects back from stack.

XML Serializer - read

```
@SuppressWarnings("unchecked")
public void read() throws Exception
{
  ObjectInputStream is = null;

  try
  {
    XStream xstream = new XStream(new DomDriver());
    is = xstream.createObjectInputStream(new FileReader(file));
    Object obj = is.readObject();
    while (obj != null)
    {
      stack.push(obj);
      obj = is.readObject();
    }
  }
  finally
  {
    if (is != null)
    {
      is.close();
    }
  }
}
```

XML Serializer - write



```
  public void write() throws Exception
  {
    ObjectOutputStream os = null;

    try
    {
      XStream xstream = new XStream(new DomDriver());
      os = xstream.createObjectOutputStream(new FileWriter(file));
      while (!stack.empty())
      {
        os.writeObject(stack.pop());
      }
    }
    finally
    {
      if (os != null)
      {
        os.close();
      }
    }
  }
}
```

Refactor PacemakerAPI to use Serializer

```
private Map<Long,   User>     userIndex       = new HashMap<>();
private Map<String, User>     emailIndex      = new HashMap<>();
private Map<Long, Activity> activitiesIndex = new HashMap<>();
```

```
public class PacemakerAPI
{
  //...

  private Serializer serializer;

  public PacemakerAPI()
  {
  }

  public PacemakerAPI(Serializer serializer)
  {
    this.serializer = serializer;
  }

  @SuppressWarnings("unchecked")
  public void load() throws Exception
  {
    serializer.read();
    activitiesIndex = (Map<Long, Activity>) serializer.pop();
    emailIndex      = (Map<String, User>)   serializer.pop();
    userIndex       = (Map<Long, User>)     serializer.pop();
  }

  void store() throws Exception
  {
    serializer.push(userIndex);
    serializer.push(emailIndex);
    serializer.push(activitiesIndex);
    serializer.write();
  }
  //...
}
```

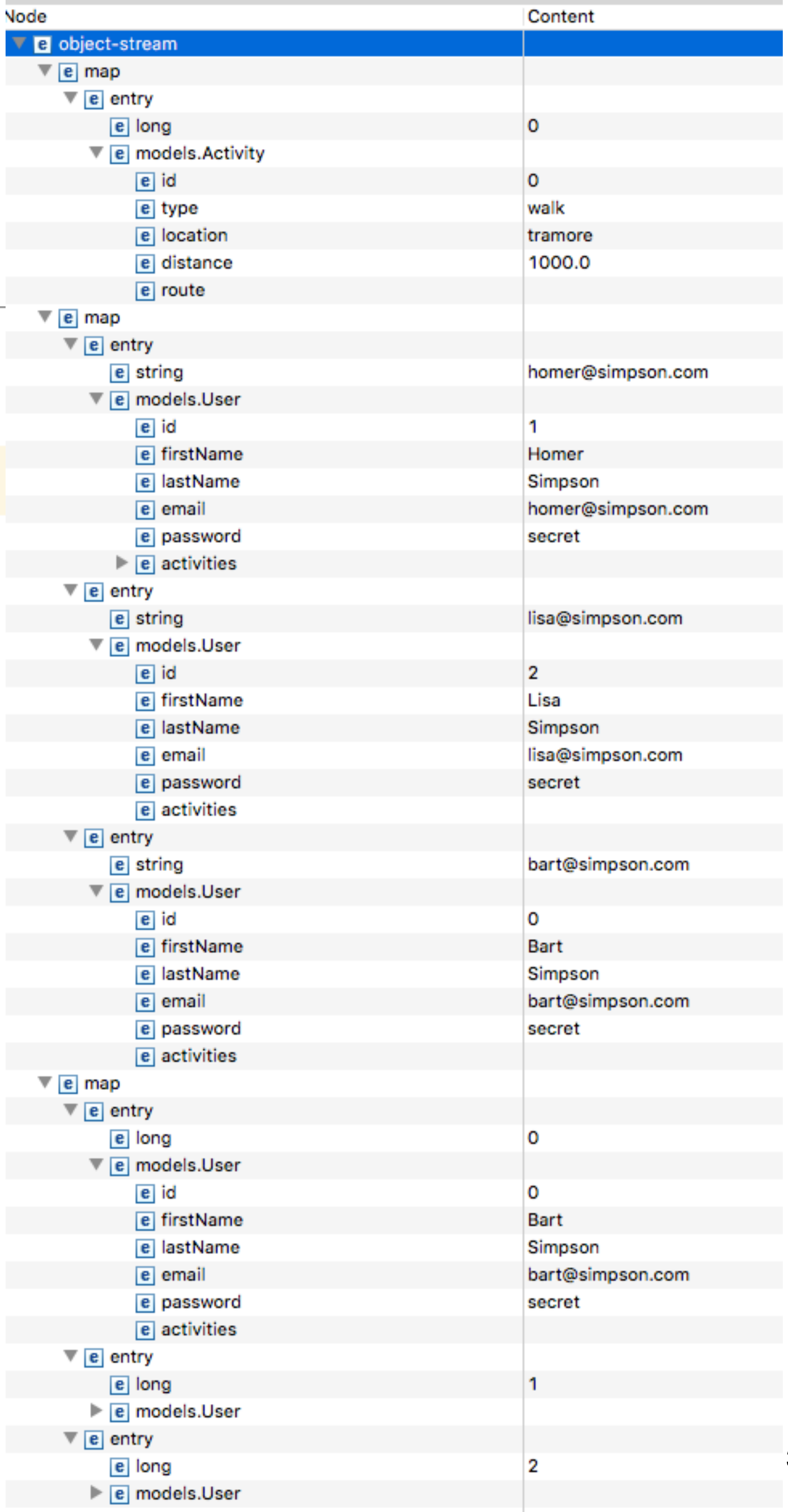
Using the Serializer

```
File  datastore = new File("datastore3.xml");
Serializer serializer = new XMLSerializer(datastore);

PacemakerAPI pacemakerAPI = new PacemakerAPI(serializer);
if (datastore.isFile())
{
  pacemakerAPI.load();
}

pacemakerAPI.createUser("Bart", "Simpson",   "bart@simpson.com", "secret");
pacemakerAPI.createUser("Homer", "Simpson",  "homer@simpson.com", "secret");
pacemakerAPI.createUser("Lisa", "Simpson", " lisa@simpson.com", "secret");

User homer = pacemakerAPI.getUserByEmail("homer@simpson.com");
pacemakerAPI.createActivity(homer.id, "walk", "tramore", 1000);

pacemakerAPI.store();
```

Sample

```
pacemakerAPI.store(new File("datastore.xml"));
```

- Stores the pacemaker model to the file ‘datastore.xml’
- The problem with the serializer is that the three Maps serialized are completely independent
- Even though the maps in memory prior to serialization contain shared

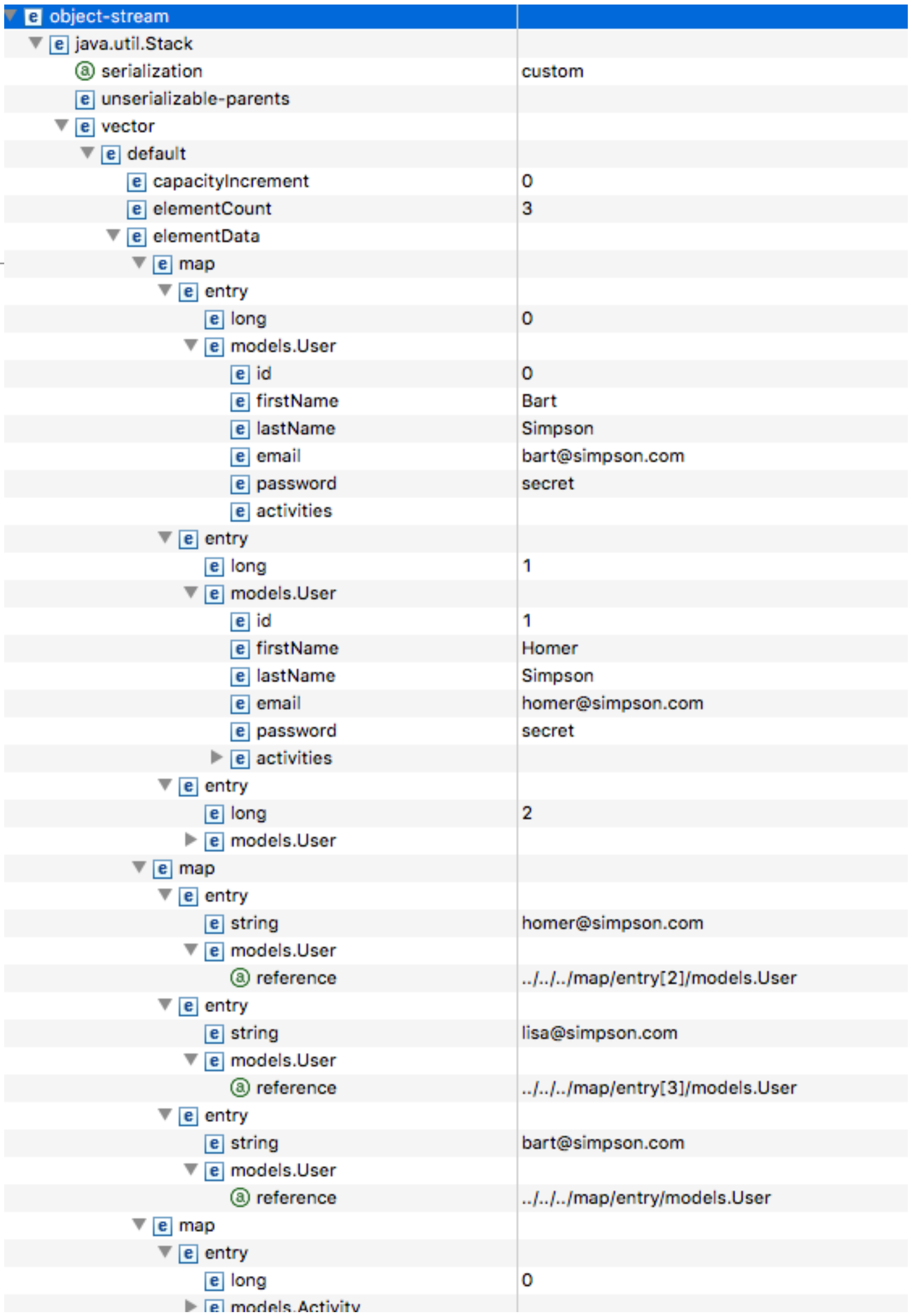


Node	Content
object-stream	
map	
entry	
long	0
models.Activity	
id	0
type	walk
location	tramore
distance	1000.0
route	
map	
entry	
string	homer@simpson.com
models.User	
id	1
firstName	Homer
lastName	Simpson
email	homer@simpson.com
password	secret
activities	
entry	
string	lisa@simpson.com
models.User	
id	2
firstName	Lisa
lastName	Simpson
email	lisa@simpson.com
password	secret
activities	
entry	
string	bart@simpson.com
models.User	
id	0
firstName	Bart
lastName	Simpson
email	bart@simpson.com
password	secret
activities	
map	
entry	
long	0
models.User	
id	0
firstName	Bart
lastName	Simpson
email	bart@simpson.com
password	secret
activities	
entry	
long	1
models.User	
entry	
long	2
models.User	

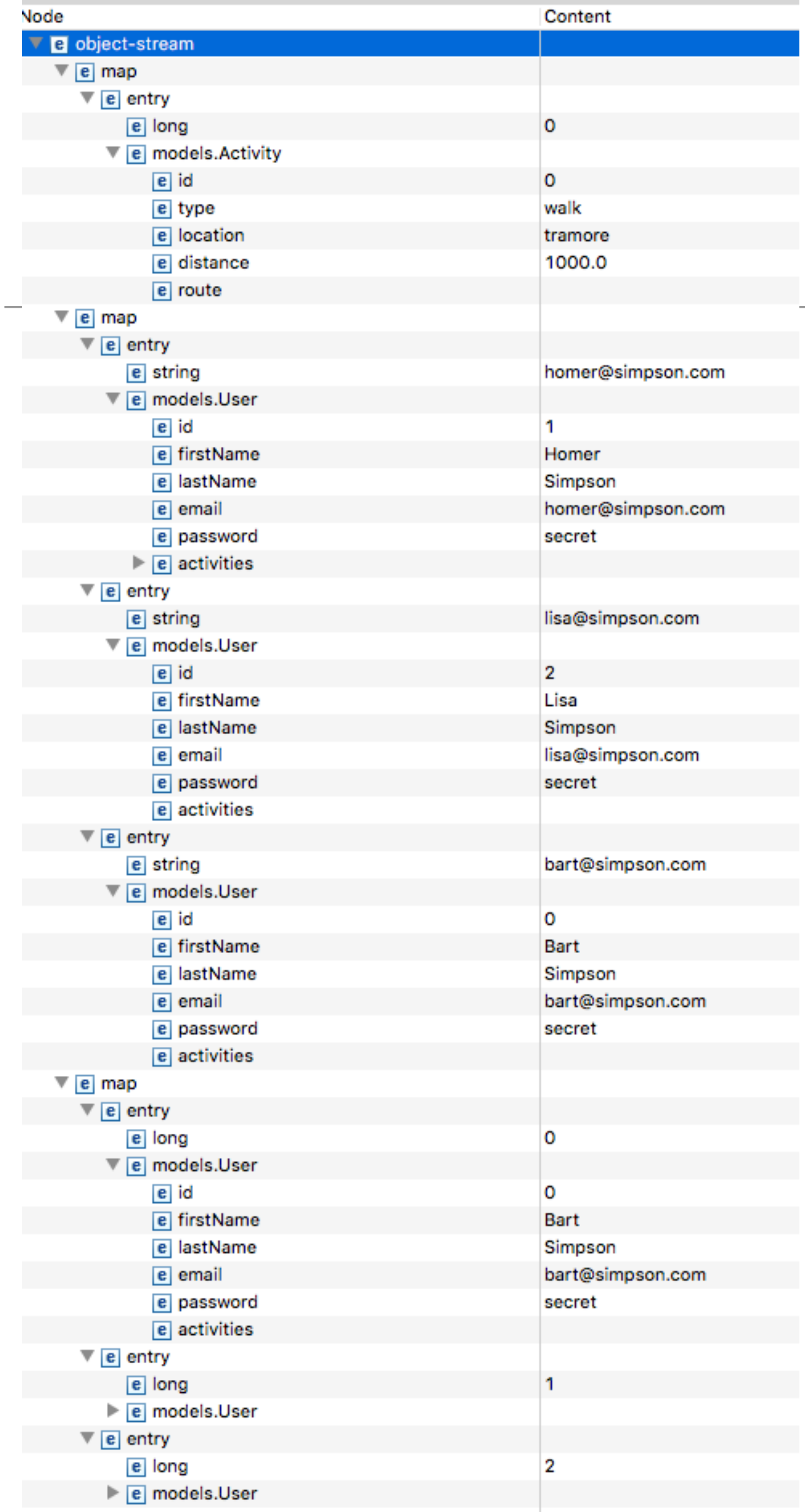
```java
@SuppressWarnings("unchecked")
public void read() throws Exception
{
  ObjectInputStream is = null;

  try
  {
    XStream xstream = new XStream(new DomDriver());
    is = xstream.createObjectInputStream(new FileReader(file));
    stack = (Stack) is.readObject();
  }
  finally
  {
    if (is != null)
    {
      is.close();
    }
  }
}

public void write() throws Exception
{
  ObjectOutputStream os = null;

  try
  {
    XStream xstream = new XStream(new DomDriver());
    os = xstream.createObjectOutputStream(new FileWriter(file));
    os.writeObject(stack);
  }
  finally
  {
    if (os != null)
    {
      os.close();
    }
  }
}
```

Object References

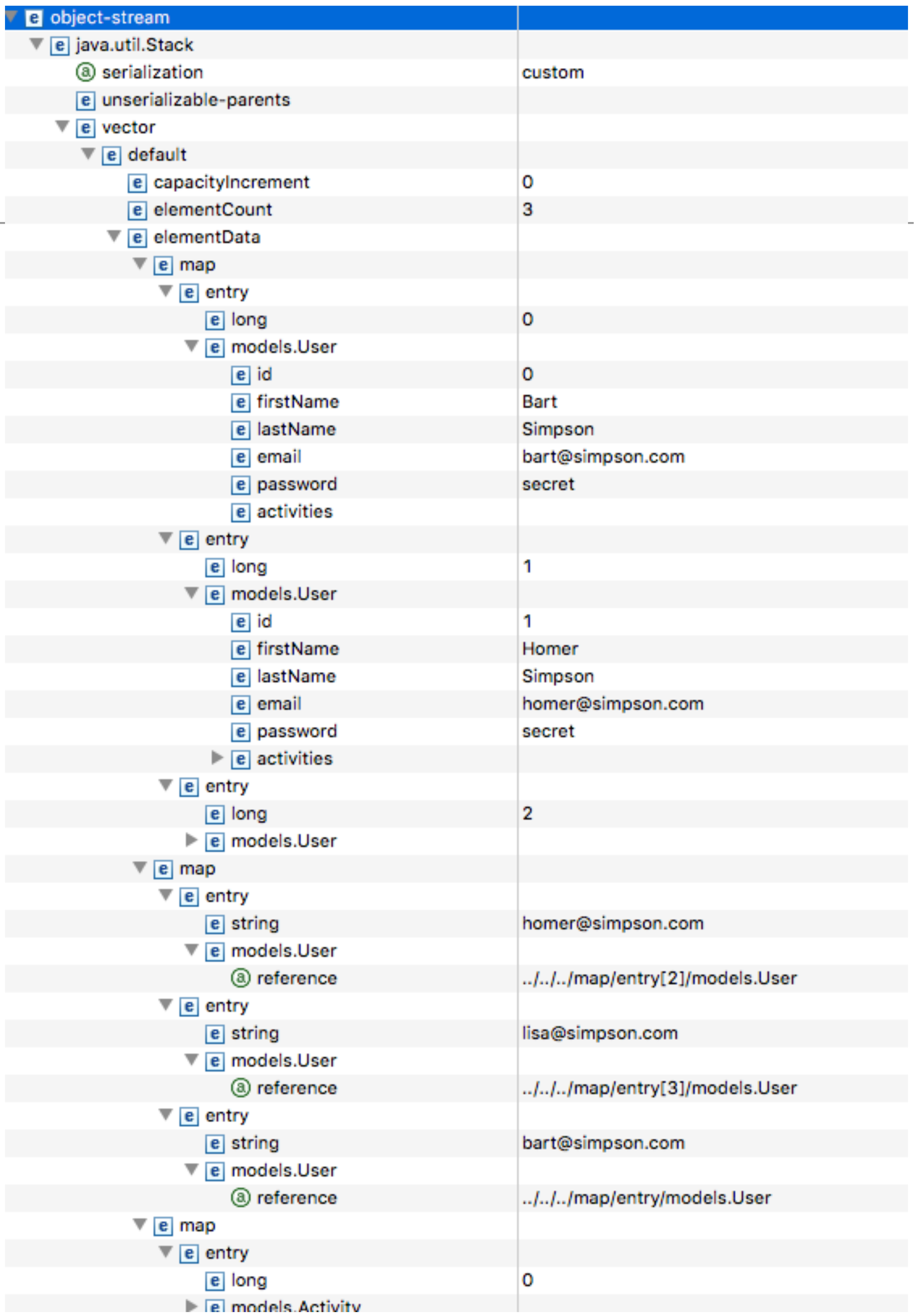
- When the objects are written - objects encountered for the second and subsequent times are detected…
- … only one copy is written.
- The second and subsequent objects are written as references..



```
▼ object-stream
  ▼ java.util.Stack
        serialization                               custom
        unserializable-parents
      ▼ vector
        ▼ default
              capacityIncrement                     0
              elementCount                          3
            ▼ elementData
              ▼ map
                ▼ entry
                      long                          0
                    ▼ models.User
                          id                        0
                          firstName                 Bart
                          lastName                  Simpson
                          email                     bart@simpson.com
                          password                  secret
                          activities
                ▼ entry
                      long                          1
                    ▼ models.User
                          id                        1
                          firstName                 Homer
                          lastName                  Simpson
                          email                     homer@simpson.com
                          password                  secret
                        ▶ activities
                ▼ entry
                      long                          2
                    ▶ models.User
              ▼ map
                ▼ entry
                      string                        homer@simpson.com
                    ▼ models.User
                          reference                 ../../../map/entry[2]/models.User
                ▼ entry
                      string                        lisa@simpson.com
                    ▼ models.User
                          reference                 ../../../map/entry[3]/models.User
                ▼ entry
                      string                        bart@simpson.com
                    ▼ models.User
                          reference                 ../../../map/entry/models.User
              ▼ map
                ▼ entry
                      long                          0
                    ▶ models.Activity
```



```
Node                                Content
▼ e object-stream
  ▼ e map
    ▼ e entry
        e long                      0
      ▼ e models.Activity
          e id                      0
          e type                    walk
          e location                tramore
          e distance                1000.0
          e route
  ▼ e map
    ▼ e entry
        e string                    homer@simpson.com
      ▼ e models.User
          e id                      1
          e firstName               Homer
          e lastName                Simpson
          e email                   homer@simpson.com
          e password                secret
        ▶ e activities
    ▼ e entry
        e string                    lisa@simpson.com
      ▼ e models.User
          e id                      2
          e firstName               Lisa
          e lastName                Simpson
          e email                   lisa@simpson.com
          e password                secret
          e activities
    ▼ e entry
        e string                    bart@simpson.com
      ▼ e models.User
          e id                      0
          e firstName               Bart
          e lastName                Simpson
          e email                   bart@simpson.com
          e password                secret
          e activities
  ▼ e map
    ▼ e entry
        e long                      0
      ▼ e models.User
          e id                      0
          e firstName               Bart
          e lastName                Simpson
          e email                   bart@simpson.com
          e password                secret
          e activities
    ▼ e entry
        e long                      1
      ▶ e models.User
    ▼ e entry
        e long                      2
      ▶ e models.User
```



```
▼ e object-stream
  ▼ e java.util.Stack
      a serialization                         custom
      e unserializable-parents
    ▼ e vector
      ▼ e default
          e capacityIncrement                 0
          e elementCount                      3
        ▼ e elementData
          ▼ e map
            ▼ e entry
                e long                        0
              ▼ e models.User
                  e id                        0
                  e firstName                 Bart
                  e lastName                  Simpson
                  e email                     bart@simpson.com
                  e password                  secret
                  e activities
            ▼ e entry
                e long                        1
              ▼ e models.User
                  e id                        1
                  e firstName                 Homer
                  e lastName                  Simpson
                  e email                     homer@simpson.com
                  e password                  secret
                ▶ e activities
            ▼ e entry
                e long                        2
              ▶ e models.User
          ▼ e map
            ▼ e entry
                e string                      homer@simpson.com
              ▼ e models.User
                  a reference                 ../../../map/entry[2]/models.User
            ▼ e entry
                e string                      lisa@simpson.com
              ▼ e models.User
                  a reference                 ../../../map/entry[3]/models.User
            ▼ e entry
                e string                      bart@simpson.com
              ▼ e models.User
                  a reference                 ../../../map/entry/models.User
          ▼ e map
            ▼ e entry
                e long                        0
              ▶ e models.Activity
```

Except where otherwise noted, this content is licensed under a Creative Commons Attribution-NonCommercial 3.0 License.

For more information, please see http://creativecommons.org/licenses/by-nc/3.0/

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning support unit