COSC 3P91

Lab 3

In this lab you are going to use object serialization, streams, and random-access files.

1. Implement a class `Person` with the following features:
   a. A person has three private fields of a `name` (type `String`), `age` (type `int`), and `occupation` (type `String`).
   b. Provide access methods for the fields. All fields can be read, and `age` and `occupation` can be set.
   c. Implement the method `toString()` so that a person with `name` "Joe Miller", `age` 32, and `occupation` "Farmer" is displayed as "Joe Miller,32,Farmer".
   d. Make the class `Serializable`.
2. In the `main` method of the application do the following:
   a. Load the strings of the file "persons.txt" and create `Person` objects based on them. Then print the persons to `System.out`. Each line of the "persons.txt" file contains the data of exactly one person.
   b. Create a file "persons.data" and save the persons that you have previously loaded in this file using object serialization.
   c. Load the persons from "person.data" back in and print them to `System.out` again.
3. Implement a class `ObjectRandomAccessFile` with the following features:
   a. An `ObjectRandomAccessFile` is a `RandomAccessFile`.
   b. The class has one constructor with parameters `name` (`String`) and `mode` (`String`) that calls the corresponding super constructor.
   c. The class has a method `writeObject(Object obj)` that writes the given object `obj` to the current position of the file if it is `Serializable`. This is done by first serializing the object in an `ObjectOutputStream` based on a `ByteArrayOutputStream`. Then you write the size (number of bytes) followed by the content of the `ByteArrayOutputStream` to the `RandomAccessFile`. Finally, the method returns the number of bytes written to the file.

   d. The class has a method `readObject()` that reads an object that is stored at the current position of the file using object deserialization. This is done by first reading the size of the serialized object. Then reading the corresponding number of bytes into a `byte[]`. Finally, create an `ObjectInputStream` based on an `ByteArrayInputStream` created of the `byte[]` and then read and return the object.
   e. The class has a method `seekObject(int pos)` that places the current I/O position to the object with index `pos` in the file. The first object has index 0.

4. Extend the `main` method by doing the following:
   a. Save the persons from above in a `ObjectRandomAccessFile` called "persons.rdata".
   b. Load the persons from the "persons.rdata" file in the following order, 3,2,4,1,5,0, i.e., read the object at index 3 first, then the object at index 2 and so on. Print the persons to `System.out`.