```
package rolodex;

import mvp.Model;
import java.io.*;

/****
 *
 * Class Card is the model class for the Card object.  A Card contains the
 * basic information for a person stored in a Rolodex: name, SSN id, age, sex,
 * and street address.
 *
 */

public class Card extends Model implements Serializable {

    /*-*
     * Public Functions
     */

    /**
     * Construct a card with the given field values.  Note that no validity
     * checking of the given values is performed here.  Rather, input value
     * checking is done in the Rolodex.Add function, q.v.
     *
     * post: (this.n == n) and (this.id == id) and (this.age == age) and
     *       (this.s == s) and (this.addr == addr);
     */
    public Card(Name n, Id id, Age age, Sex s, Address addr) {
        super(null);
        this.n = n;
        this.id = id;
        this.age = age;
        this.s = s;
        this.addr = addr;
    }


    /*-*
     * Field access  functions.
     */

    /**
     * Return the name of this.
     *                                                          <pre>
     * post: return == n;
     *                                                          </pre>
     */
    public Name getName() {
        return n;
    }

    /**
     * Return the id of this.
     *                                                          <pre>
     * post: return == id;
     *                                                          </pre>
     */
    public Id getId() {
        return id;
    }

    /**
     * Return the age of this.
     *
     * post: return == age;
     */
    public Age getAge() {
        return age;
    }

    /**
     * Return the sex of this.
     *
     * post: return == s;
     */
    public Sex getSex() {
        return s;
    }

    /**
     * Return the name of this.
     *
     * post: return == addr;
     */
    public Address getAddr() {
        return addr;
    }

    /*
     * Return true if this is deep equal to the given card , i.e. card all of
     * the fields of this are equal to all of the fields of the given card.
     *
     * post: (this.n == c.n) and (this.id == c.id) and
     *       (this.age == c.age) and (this.s == c.s) and
     *       (this.addr == c.addr);
     */
    public boolean equals(Card c) {
        return
            c.n.equals(n) &&
            c.id.equals(id) &&
            c.age.equals(age) &&
            c.s.equals(s) &&
            c.addr.equals(addr);
    }

    /*
     * Print this to stdout.  In the current rolodex design, the specs for
     * print are the same as for dump below, q.v.  In general, if there is a
     * distinction between print and dump it is that the former produces a more
     * nicely formated output than the latter.
     */
    public void print() {
```

```
113         dump();
114     }
115
116     /**
117      * Dump this to stdout.
118      *
119      * post: (stdout == "Name: " + n + "\n" +
120      *                  "Id: " + id + "\n" +
121      *                  "Age: " + age + "\n" +
122      *                  "Sex: " + s + "\n" +
123      *                  "Addr: " + addr + "\n\n")
124      */
125     public void dump() {
126         System.out.print(
127             "Name: " + n.stringValue() + "\n" +
128             "Id: " + id.integerValue().toString() + "\n" +
129             "Age: " + age.integerValue().toString() + "\n" +
130             "Sex: " + s.stringValue() + "\n" +
131             "Addr: " + addr.stringValue() + "\n\n");
132     }
133
134
135     /*-*
136      * Data Members
137      */
138
139     /** Person name and record search key */
140     protected Name n;
141
142     /** Person id and unique record key */
143     protected Id id;
144
145     /** Person age */
146     protected Age age;
147
148     /** Person sex */
149     protected Sex s;
150
151     /** Person street address */
152     protected Address addr;
153
154 }
```