```java
package project4;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.util.ArrayList;

import javax.swing.AbstractListModel;

/*******************************************************************
 * rental store that holds the list of rented DVDs,
 * used in RentalStoreGUI
 * @author Gregory Huizenga
 * @version 7/26/2017
 *******************************************************************/
public class RentalStore extends AbstractListModel implements Serializable{

    /** the list containing the rented dvds */
    private MyDoubleLinkedList<DVD> storeList;

    /*******************************************************************
     * default constructor for RentalStore
     *****************************************************************/
    public RentalStore(){
        storeList = new MyDoubleLinkedList<DVD>();
    }

    /*******************************************************************
     * add method for dvds
     * @param item the dvd to be added
     *****************************************************************/
    public void add(DVD item){
        storeList.add(item);
    }

    /*******************************************************************
     * add method for blue rays
     * @param item the blue ray to be added
     *****************************************************************/
    public void add(BlueRay item){
        storeList.add(item);
    }

    /*******************************************************************
     * add method for games
     * @param item the game to be added
     *****************************************************************/
    public void add(Game item){
        storeList.add(item);
    }

    /*******************************************************************
     * removes the item at the specified index from the list
     * @param index the index of the item to be removed
     * @return temp the element that was removed
     *************************************************************/
    public DVD remove(int index){
        DVD temp = (DVD) storeList.get(index);
        storeList.remove(index);
        return temp;
    }

    @Override
    /**************************************************************
     * returns the element stored at the given index
     * @return index index of item
     *************************************************************/
    public Object getElementAt(int index) {
        return storeList.get(index);
    }

    @Override
    /**************************************************************
     * returns the current size of the list
     * @return size size of list
     *************************************************************/
    public int getSize() {
        return storeList.size();
    }

    /**************************************************************
     * updates the jlist in the gui
     *************************************************************/
    public void update(){
        this.fireContentsChanged(this, 0, storeList.size() - 1);
    }

    /**************************************************************
     * saves the current list to a file with specified name
     * @param filename the name to be saved to
     *************************************************************/
    public void saveToFile(String filename){
        try {
            FileOutputStream outputF = new FileOutputStream(filename);
            ObjectOutputStream outputO = new ObjectOutputStream(outputF);

            outputO.writeObject(storeList);
            outputO.close();

        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    /**************************************************************
     * loads to the current list from a file with specified name
     * @param filename the name to be loaded from
     *************************************************************/
```

```java
    public void loadFromFile(String filename){
        try {
            FileInputStream inputF = new FileInputStream(filename);
            ObjectInputStream inputO = new ObjectInputStream(inputF);

            storeList = (MyDoubleLinkedList<DVD>) inputO.readObject();
        }catch(IOException e){
            e.printStackTrace();
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }

}

```