Esercizio “Biblioteca”

Implementare un programma in grado di tenere traccia dei libri presenti in una biblioteca.
Il programma deve poter:

1) Stampare una lista di tutti i libri presenti nella biblioteca e del loro stato (prestato/presente);
2) Effettuare ricerca con chiave uguale al titolo e stampare il risultato come sopra;
3) Effettuare ricerca con chiave uguale all’autore e stampare il risultato come sopra;
4) Effettuare ricerca con chiavi uguali a titolo e autore e stampare il risultato come sopra;
5) Dato un titolo e un autore aggiungere un libro alla biblioteca corrente;
6) Dato un titolo e un autore prestare il libro corrispondente e tenere traccia del possessore corrente;
7) Dato un titolo e un autore accettare prenotazioni per il libro corrispondente e tenere per ogni libro una lista delle prenotazioni.

Il programma deve quindi fornire un prompt e accettare i comandi corrispondenti alle varie opzioni. A seguito del comando selezionato (es. add/search/borrow/reserve) il programma richiedera’ in sequenza i dati necessari ed effettuera’ l’operazione richiesta.
Il tentativo di effettuare un’operazione su un libr inesistente deve generare un messaggio di errore e NON abortire il programma.

Il programma deve essere in grado di mantenere uno stato su memoria non volatile (disco). A tal fine la funzione main accettera’ da linea di comando un nome di file. Il file cosi’ nominato identifica la copia stabile dello stato del sistema. Nel caso il file identificato sia assente si assume lo stato iniziale di biblioteca vuota.

Per salvare e recuperare lo stato e per leggere dati da linea di comando si utilizzeranno le funzioni della classe Utility di seguito listata.

```
import java.io.*;

public class Utility
{
	private static BufferedReader br = null;

	public static String readInputLine()
	{
		String retval = null;
		try
		{
			retval = br.readLine();
		}
		catch(IOException ioe)
		{
			ioe.printStackTrace();
		}
		return retval;
	}

	public static boolean storeStatus(Object o, String nomeFile)
	{
		boolean retval = false;
		try
		{
			FileOutputStream fos = new FileOutputStream(nomeFile);
			ObjectOutputStream oos = new ObjectOutputStream(fos);
			oos.writeObject(o);
			oos.close();
			fos.close();
			retval = true;
		}
		catch(IOException ioe)
		{
			ioe.printStackTrace();
		}
		return retval;
	}

	public static Object recoverStatus(String nomeFile)
	{
		Object retval = null;
		try
		{
			FileInputStream fis = new FileInputStream(nomeFile);
			ObjectInputStream ois = new ObjectInputStream(fis);
			retval = ois.readObject();
			ois.close();
			fis.close();
		}
		catch(ClassNotFoundException cnfe)
		{
			cnfe.printStackTrace();
		}
		catch(IOException ioe)
		{
			ioe.printStackTrace();
		}
		return retval;
	}

	static
	{
		br = new BufferedReader(new InputStreamReader(System.in));
	}
}
```