

Crie uma situação em que ocorra uma exceção dentro de um código. Utilize try/catch para realizar a captura e tratamento dessa exceção.

```
import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;

public class Empacotamento {

    // serialização: gravando o objetos no arquivo binário "nomeArq"
    public static void gravarArquivoBinario(ArrayList<Object> lista, String nomeArq) {
        File arq = new File(nomeArq);
        try {
            arq.delete();
            arq.createNewFile();

            ObjectOutputStream objOutput = new ObjectOutputStream(new FileOutputStream(arq));
            objOutput.writeObject(lista);
            objOutput.close();

        } catch(IOException erro) {
            System.out.printf("Erro: %s", erro.getMessage());
        }
    }

    // desserialização: recuperando os objetos gravados no arquivo binário "nomeArq"
    public static ArrayList<Object> lerArquivoBinario(String nomeArq) {
        ArrayList<Object> lista = new ArrayList();
        try {
            File arq = new File(nomeArq);
            if (arq.exists()) {
                ObjectInputStream objInput = new ObjectInputStream(new FileInputStream(arq));
                lista = (ArrayList<Object>)objInput.readObject();
                objInput.close();
            }
        } catch(IOException erro1) {
            System.out.printf("Erro: %s", erro1.getMessage());
        } catch(ClassNotFoundException erro2) {
            System.out.printf("Erro: %s", erro2.getMessage());
        }

        return(lista);
    }
}
```