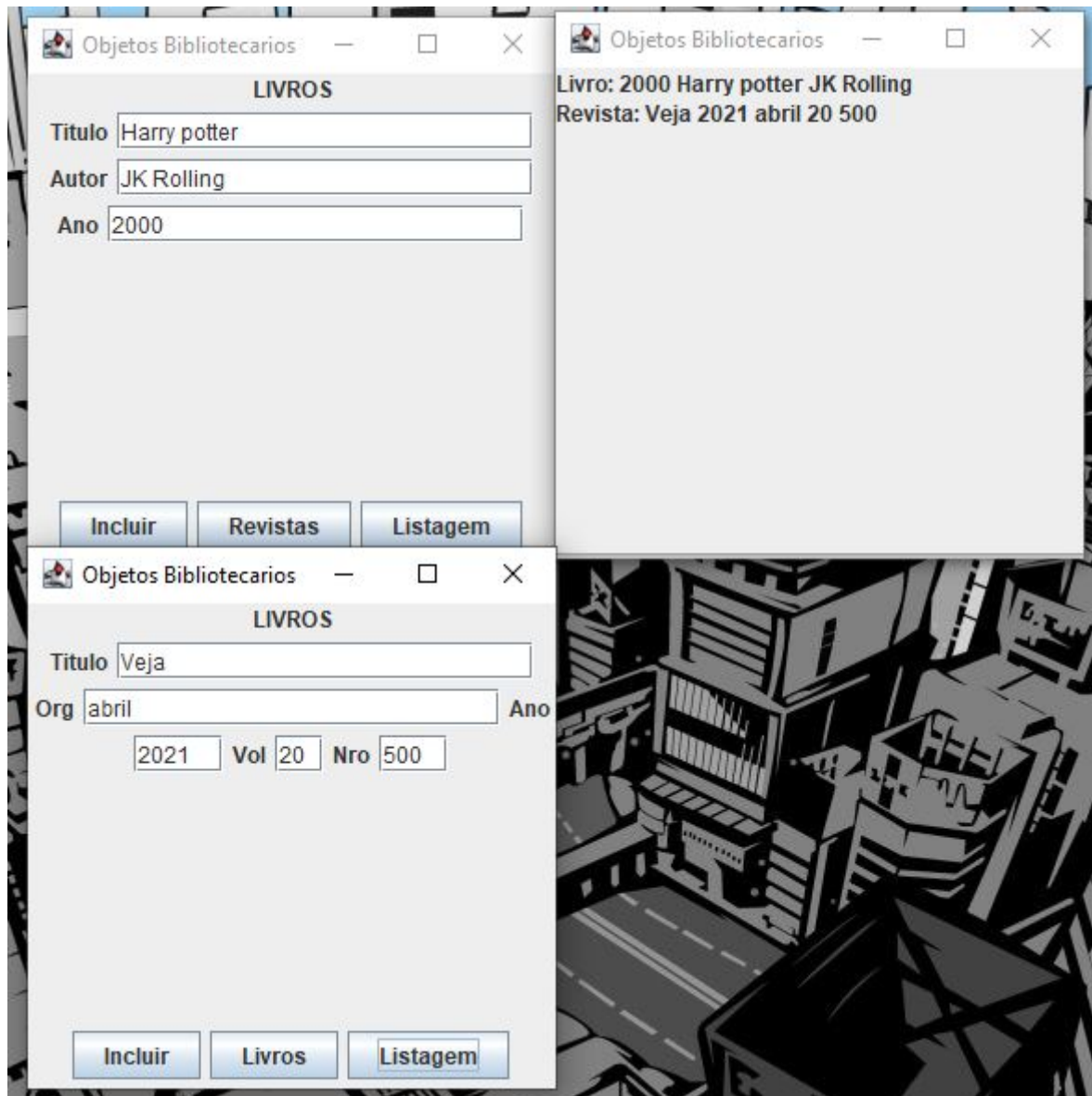


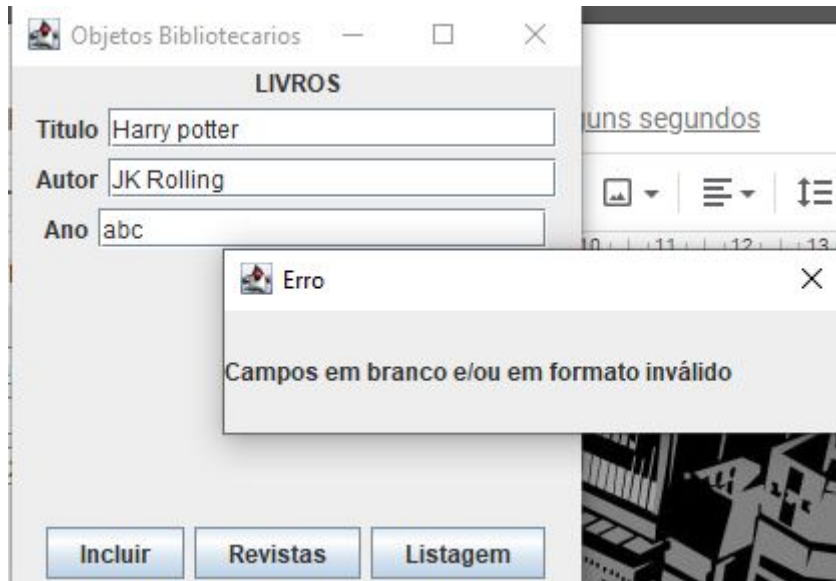
Nessa prática tentei ao máximo executar os princípios do SOLID, creio que consegui manter todas as práticas dele. Alguns code smells que não consegui evitar nas classes que desenham as interfaces foram a “Large class” e o “Duplicated code”, provavelmente existem outros smells no meu código já que ainda estou aprendendo a tomar essas precauções, porém fiz o máximo para não ocorrerem.

### Execução do programa:

Sem nenhum erro



Execução com erro(Letra no lugar de número):



Código da Aplicação:

app.java

```
import java.util.ArrayList;
import java.util.List;

public class App {
    public static void main(String[] args) throws Exception {
        List<ObjBibliotecario> objetos = new
ArrayList<ObjBibliotecario>();

        LivrosGUI livrosGUI = new LivrosGUI(objetos);

        livrosGUI.startGUI();
    }
}
```

ObjBibliotecario.java

```
import java.util.List;

public abstract class ObjBibliotecario implements ObjBiblioInterface{
    private String titulo;
    private int ano;

    ObjBibliotecario(String tituloIn,int anoIn) throws Exception{
        titulo = tituloIn;
    }
}
```

```

        if(titulo.isBlank()){
            throw new Exception("Null Field");
        }

        try{
            ano = anoIn;
        }
        catch(NumberFormatException e){
            throw new Exception("Not a number");
        }
    }

    public void incluir(List<ObjBibliotecario> dados){
        dados.add(this);
    }

    public int getAno() {
        return ano;
    }

    public String getTitulo() {
        return titulo;
    }

    public String print(){
        return ano+" "+titulo;
    }
}

```

---

### Livro.java

```

public class Livro extends ObjBibliotecario{
    protected String autor;

    Livro(String titulo, int ano, String autorIn) throws Exception{
        super(titulo, ano);
        autor = autorIn;
        if(autorIn.isBlank()){
            throw new Exception("Null Field");
        }
    }

    public String getAutor() {

```

```

        return autor;
    }

    @Override
    public String print() {
        return "Livro: " +super.print() + " " + autor;
    }
}

```

## Revista.java

```

public class Revista extends ObjBibliotecario{
    protected String org;
    protected int vol;
    protected int nro;

    Revista(String titulo, int ano, String orgIn, int volIn, int nroIn)
throws Exception{
        super(titulo,ano);
        org = orgIn;
        if(orgIn.isBlank()){
            throw new Exception("Null Field");
        }
        try{
            vol = volIn;
            nro = nroIn;
        }
        catch(NumberFormatException e){
            throw new Exception("Not a number field");
        }
    }

    public int getNro() {
        return nro;
    }

    public String getOrg() {
        return org;
    }

    public int getVol() {
        return vol;
    }
}

```

```

        @Override
        public String print(){
            return "Revista: " + this.getTitulo() + " " + this.getAno() + "
"+ org + " " + vol + " " + nro;
        }
    }
}

```

---

### ObjBiblioInterface.java

```

interface ObjBiblioInterface {
    public String print();
}

```

---

### LivrosGUI.java

```

import javax.swing.*;

import static javax.swing.WindowConstants.EXIT_ON_CLOSE;

import java.awt.*;
import java.awt.event.*;
import java.util.List;

public class LivrosGUI implements ActionListener{
    private List <ObjBibliotecario> vetor;
    //Creating JFrame
    private JFrame frame = new JFrame("Objetos Bibliotecarios");
    //Creating Buttons
    private JButton incluirButton = new JButton("Incluir");
    private JButton revistasButton = new JButton("Revistas");
    private JButton listagemButton = new JButton("Listagem");
    //Creating JPanels
    private JPanel buttonPanel = new JPanel();
    private JPanel tituloPanel = new JPanel();
    //Creating Labels
    private JLabel tituloLabel = new JLabel("LIVROS",
SwingConstants.CENTER);
    private JLabel autorLabel = new JLabel("Autor");
    private JLabel livroTituloLabel = new JLabel("Titulo");
    private JLabel anoLabel = new JLabel("Ano",
SwingConstants.CENTER);
    //Creating Text Fields

```

```

private JTextField tituloTf = new JTextField(20);
private JTextField autorTf = new JTextField(20);
private JTextField anoTf = new JTextField(20);

LivrosGUI(List<ObjBibliotecario> lista){
    vetor = lista;
}

public void startGUI(){
    frame.setDefaultCloseOperation(EXIT_ON_CLOSE);
    frame.setSize(300,300);

    tituloPanel.add(livroTituloLabel);
    tituloPanel.add(tituloTf);
    tituloPanel.add(autorLabel);
    tituloPanel.add(autorTf);
    tituloPanel.add(anoLabel);
    tituloPanel.add(anoTf);

    buttonPanel.add(incluirButton);
    buttonPanel.add(revistasButton);
    buttonPanel.add(listagemButton);

    incluirButton.addActionListener(this);
    revistasButton.addActionListener(this);
    listagemButton.addActionListener(this);

    frame.getContentPane().add(BorderLayout.SOUTH, buttonPanel);
    frame.getContentPane().add(BorderLayout.PAGE_START,
tituloLabel);
    frame.getContentPane().add(BorderLayout.CENTER, tituloPanel);

    frame.setVisible(true);
}

@Override
public void actionPerformed(ActionEvent e) {
    if(e.getActionCommand().equals("Incluir")){
        try{
            vetor.add(new Livro(tituloTf.getText(),
Integer.parseInt(anoTf.getText()), autorTf.getText()));
        }
        catch(Exception exc){

```

```

        JDialog dialog = new JDialog(frame, "Erro");

        JLabel l = new JLabel("Campos em branco e/ou em formato
inválido");

        dialog.add(l);
        dialog.setSize(100, 100);
        dialog.setVisible(true);
    }
}

if(e.getActionCommand().equals("Listagem")){
    ListaObjetosGUI lista = new ListaObjetosGUI();
    lista.printObj(vetor);
}

if(e.getActionCommand().equals("Revistas")){
    RevistasGUI revista = new RevistasGUI(vetor);
    revista.startGUI();
}
}
}

```

---

### RevistasGUI.java

```

import javax.swing.*;
import java.util.List;
import java.awt.event.*;
import static javax.swing.WindowConstants.EXIT_ON_CLOSE;

import java.awt.*;

public class RevistasGUI implements ActionListener{
    private List <ObjBibliotecario> vetor;
    //Creating JFrame
    private JFrame frame = new JFrame("Objetos Bibliotecarios");
    //Creating Buttons
    private JButton incluirButton = new JButton("Incluir");
    private JButton revistasButton = new JButton("Livros");
    private JButton listagemButton = new JButton("Listagem");
    //Creating JPanels
    private JPanel buttonPanel = new JPanel();
    private JPanel tituloPanel = new JPanel();
}

```

```

//Creating Labels
private JLabel tituloLabel = new JLabel("LIVROS",
SwingConstants.CENTER);

private JLabel orgLabel = new JLabel("Org");
private JLabel revistaTituloLabel = new JLabel("Titulo");
private JLabel anoLabel = new JLabel("Ano");
private JLabel volLabel = new JLabel("Vol");
private JLabel nroLabel = new JLabel("Nro");

//Creating Text Fields
private JTextField tituloTf = new JTextField(20);
private JTextField orgTf = new JTextField(20);
private JTextField anoTf = new JTextField(4);
private JTextField volTf = new JTextField(2);
private JTextField nroTf = new JTextField(3);

RevistasGUI(List <ObjBibliotecario> vetorIn){
    vetor = vetorIn;
}

public void startGUI(){
    frame.setDefaultCloseOperation(EXIT_ON_CLOSE);
    frame.setSize(300,300);

    tituloPanel.add(revistaTituloLabel);
    tituloPanel.add(tituloTf);
    tituloPanel.add(orgLabel);
    tituloPanel.add(orgTf);
    tituloPanel.add(anoLabel);
    tituloPanel.add(anoTf);
    tituloPanel.add(volLabel);
    tituloPanel.add(volTf);
    tituloPanel.add(nroLabel);
    tituloPanel.add(nroTf);

    buttonPanel.add(incluirButton);
    buttonPanel.add(revistasButton);
    buttonPanel.add(listagemButton);

    incluirButton.addActionListener(this);
    revistasButton.addActionListener(this);
    listagemButton.addActionListener(this);

    frame.getContentPane().add(BorderLayout.SOUTH, buttonPanel);

```



```

        frame.getContentPane().add(BorderLayout.PAGE_START,
tituloLabel);
        frame.getContentPane().add(BorderLayout.CENTER, tituloPanel);

        frame.setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        if(e.getActionCommand().equals("Incluir")){
            try{
                vetor.add(new Revista(tituloTf.getText(),
Integer.parseInt(anoTf.getText()), orgTf.getText(),
Integer.parseInt(volTf.getText()), Integer.parseInt(nroTf.getText())));
            }
            catch(Exception exc){
                JDialog dialog = new JDialog(frame,"Erro");

                JLabel l = new JLabel("Campos em branco e/ou em formato
inválido");

                dialog.add(l);
                dialog.setSize(100,100);
                dialog.setVisible(true);
            }
        }

        if(e.getActionCommand().equals("Listagem")){
            ListaObjetosGUI lista = new ListaObjetosGUI();
            lista.printObj(vetor);
        }

        if(e.getActionCommand().equals("Livros")){
            LivrosGUI livro = new LivrosGUI(vetor);
            livro.startGUI();
        }
    }
}

```

---

### ListaObjetosGUI.java

```

import java.util.ArrayList;
import java.util.List;
import static javax.swing.WindowConstants.EXIT_ON_CLOSE;

```

```

import javax.swing.*;

public class ListaObjetosGUI {

    private JFrame frame = new JFrame("Objetos Bibliotecarios");

    private JPanel printPanel = new JPanel();

    private List<JLabel> labels = new ArrayList<>();

    public void printObj(List<ObjBibliotecario> lista) {
        frame.setDefaultCloseOperation(EXIT_ON_CLOSE);
        frame.setSize(300,300);

        try{
            printPanel.setLayout(new BoxLayout(printPanel,
BoxLayout.Y_AXIS));
            for(ObjBibliotecario o:lista){
                String dados = o.print();
                labels.add(new JLabel (dados));
            }

            for (JLabel j:labels){
                printPanel.add(j);
            }

            frame.getContentPane().add(printPanel);
            frame.setVisible(true);
        }
        catch(Exception e){
            JDialog dialog = new JDialog(frame,"Erro");

            JLabel l = new JLabel("Algum erro inesperado aconteceu:\n"
+ e.toString());

            dialog.add(l);
            dialog.setSize(100,100);
            dialog.setVisible(true);
        }
    }
}

```

