

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Paulo Kiyoshi Oyama Filho

Prática 01 - Revisão POO

O código implementado tenta seguir o padrão MVC (Model, View e Controller).

1.Código

a. Model

```
public abstract class BaseType {

    public String title;
    public int year;

    public BaseType(String title, int year) {
        this.title = title;
        this.year = year;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public int getYear() {
        return year;
    }

    public void setYear(int year) {
        this.year = year;
    }

    public String print() {
        return title + " " + year + "\n";
    }

}

public class Livros extends BaseType {
```

```

    public String author;

    public Livros(String title, int year, String author) {
        super(title, year);
        this.author = author;
    }

    public String getAuthor() {
        return author;
    }

    public void setAuthor(String autor) {
        this.author = autor;
    }

    @Override
    public String print() {
        return "Livros: " + super.title + " " + author + " " +
super.year + "\n";
    }
}

public class Revista extends BaseType {

    public String org;
    public int volume;
    public int number;

    public Revista(String org, int volume, int number, String
title, int year) {
        super(title, year);
        this.org = org;
        this.volume = volume;
        this.number = number;
    }

    public String getOrg() {
        return org;
    }
}

```

```

    public void setOrg(String org) {
        this.org = org;
    }

    public int getVolume() {
        return volume;
    }

    public void setVolume(int volume) {
        this.volume = volume;
    }

    public int getNumber() {
        return number;
    }

    public void setNumber(int number) {
        this.number = number;
    }

    @Override
    public String print() {
        return "Revista: " + super.title + " " + org + " " + volume +
            " " + number + " " + super.year + "\n";
    }
}

public class Listagem {

    public List<BaseType> library;

    public Listagem() {
        this.library = new ArrayList<BaseType>();
    }
}

```

b. View

```

public class LivroForm
    extends JFrame {

    final protected Container c;
    final public JLabel typeLabel, authorLabel, titleLabel,
yearLabel;
    final public JTextField titleTextField, yearTextField,
authorTextField;
    final public JButton btnAdd, btnType, btnList;
    final LivrosController controller = new LivrosController();

    public LivroForm(String title, String geral) {
        setBounds(300, 90, 900, 600);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setResizable(false);

        c = getContentPane();
        c.setLayout(null);

        typeLabel = new JLabel(title);
        typeLabel.setFont(new Font("Arial", Font.PLAIN, 30));
        typeLabel.setSize(300, 30);
        typeLabel.setLocation(220, 30);
        c.add(typeLabel);

        //FIRST ROW
        titleLabel = new JLabel("Titulo");
        titleLabel.setFont(new Font("Arial", Font.PLAIN, 20));
        titleLabel.setSize(100, 20);
        titleLabel.setLocation(100, 100);
        c.add(titleLabel);

        titleTextField = new JTextField();
        titleTextField.setFont(new Font("Arial", Font.PLAIN, 15));
        titleTextField.setSize(190, 20);
        titleTextField.setLocation(200, 100);
        c.add(titleTextField);

        //SECOND ROW
        authorLabel = new JLabel(geral);
        authorLabel.setFont(new Font("Arial", Font.PLAIN, 20));

```

```

authorLabel.setSize(100, 20);
authorLabel.setLocation(100, 150);
c.add(authorLabel);

authorTextField = new JTextField();
authorTextField.setFont(new Font("Arial", Font.PLAIN, 15));
authorTextField.setSize(190, 20);
authorTextField.setLocation(200, 150);
c.add(authorTextField);

//THIRD ROW
yearLabel = new JLabel("Ano");
yearLabel.setFont(new Font("Arial", Font.PLAIN, 20));
yearLabel.setSize(100, 20);
yearLabel.setLocation(100, 200);
c.add(yearLabel);

yearTextField = new JTextField();
yearTextField.setFont(new Font("Arial", Font.PLAIN, 15));
yearTextField.setSize(50, 20);
yearTextField.setLocation(140, 200);
c.add(yearTextField);

//FORTH ROW
btnAdd = new JButton("Incluir");
btnAdd.setFont(new Font("Arial", Font.PLAIN, 15));
btnAdd.setSize(100, 20);
btnAdd.setLocation(100, 250);
btnAdd.addActionListener((ActionEvent e) -> {
    int year = Integer.parseInt(yearTextField.getText());
    controller.isFieldsNotNull(titleTextField.getText(),
year, authorTextField.getText());

    BaseType bt = new Livros(titleTextField.getText(), year,
authorTextField.getText());
    controller.include(titleTextField.getText(), year,
authorTextField.getText());
});
c.add(btnAdd);

btnType = new JButton("Revistas");
btnType.setFont(new Font("Arial", Font.PLAIN, 15));

```

```

        btnType.setSize(100, 20);
        btnType.setLocation(220, 250);
        btnType.addActionListener((ActionEvent e) -> {
            setVisible(false);
            RevistaForm re = new RevistaForm("Revista", "Org.");
        });
        c.add(btnType);

        btnList = new JButton("Listagem");
        btnList.setFont(new Font("Arial", Font.PLAIN, 15));
        btnList.setSize(100, 20);
        btnList.setLocation(340, 250);
        btnList.addActionListener((ActionEvent e) -> {
            setVisible(false);
            ListagemView li = new ListagemView(controller.toList());
        });
        c.add(btnList);

        setVisible(true);
    }
}

```

```

public class RevistaForm extends JFrame {

    final protected Container c;
    final public JLabel typelabel, geralLabel, titleLabel,
yearLabel;
    final public JTextField titleTextField, yearTextField,
geralTextField;
    final public JButton btnAdd, btnType, btnList;
    final private JLabel numberLabel, volumeLabel;
    final private JTextField numberTextField, volumeTextField;
    final RevistaController controller = new RevistaController();

    public RevistaForm(String title, String geral) {

        setBounds(300, 90, 900, 600);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setResizable(false);
    }
}

```

```

c = getContentPane();
c.setLayout(null);

titleLabel = new JLabel(title);
titleLabel.setFont(new Font("Arial", Font.PLAIN, 30));
titleLabel.setSize(300, 30);
titleLabel.setLocation(220, 30);
c.add(titleLabel);

//FIRST ROW
titleLabel = new JLabel("Titulo");
titleLabel.setFont(new Font("Arial", Font.PLAIN, 20));
titleLabel.setSize(100, 20);
titleLabel.setLocation(100, 100);
c.add(titleLabel);

titleTextField = new JTextField();
titleTextField.setFont(new Font("Arial", Font.PLAIN, 15));
titleTextField.setSize(190, 20);
titleTextField.setLocation(200, 100);
c.add(titleTextField);

//SECOND ROW
geralLabel = new JLabel(geral);
geralLabel.setFont(new Font("Arial", Font.PLAIN, 20));
geralLabel.setSize(100, 20);
geralLabel.setLocation(100, 150);
c.add(geralLabel);

geralTextField = new JTextField();
geralTextField.setFont(new Font("Arial", Font.PLAIN, 15));
geralTextField.setSize(190, 20);
geralTextField.setLocation(200, 150);
c.add(geralTextField);

//THIRD ROW
yearLabel = new JLabel("Ano");
yearLabel.setFont(new Font("Arial", Font.PLAIN, 20));
yearLabel.setSize(100, 20);
yearLabel.setLocation(100, 200);
c.add(yearLabel);

```



```

yearTextField = new JTextField();
yearTextField.setFont(new Font("Arial", Font.PLAIN, 15));
yearTextField.setSize(50, 20);
yearTextField.setLocation(140, 200);
c.add(yearTextField);

numberLabel = new JLabel("Nro.");
numberLabel.setFont(new Font("Arial", Font.PLAIN, 20));
numberLabel.setSize(100, 20);
numberLabel.setLocation(220, 200);
c.add(numberLabel);

numberTextField = new JTextField();
numberTextField.setFont(new Font("Arial", Font.PLAIN, 15));
numberTextField.setSize(50, 20);
numberTextField.setLocation(260, 200);
c.add(numberTextField);

volumeLabel = new JLabel("Vol.");
volumeLabel.setFont(new Font("Arial", Font.PLAIN, 20));
volumeLabel.setSize(100, 20);
volumeLabel.setLocation(340, 200);
c.add(volumeLabel);

volumeTextField = new JTextField();
volumeTextField.setFont(new Font("Arial", Font.PLAIN, 15));
volumeTextField.setSize(50, 20);
volumeTextField.setLocation(380, 200);
c.add(volumeTextField);

//FORTH ROW
btnAdd = new JButton("Incluire");
btnAdd.setFont(new Font("Arial", Font.PLAIN, 15));
btnAdd.setSize(100, 20);
btnAdd.setLocation(100, 250);
btnAdd.addActionListener((ActionEvent e) -> {
    int year = Integer.parseInt(yearTextField.getText());
    int volume =
Integer.parseInt(volumeTextField.getText());
    int number =
Integer.parseInt(numberTextField.getText());

```

```

        controller.isFieldsNotNull(geralTextField.getText(),
volume, number, titleTextField.getText(), year);

        controller.include(geralTextField.getText(), volume,
number, titleTextField.getText(), year);
    });
    c.add(btnAdd);

    btnType = new JButton("Livros");
    btnType.setFont(new Font("Arial", Font.PLAIN, 15));
    btnType.setSize(100, 20);
    btnType.setLocation(220, 250);
    btnType.addActionListener((ActionEvent e) -> {
        setVisible(false);
        LivroForm re = new LivroForm("Livros", "Autor");
    });
    c.add(btnType);

    btnList = new JButton("Listagem");
    btnList.setFont(new Font("Arial", Font.PLAIN, 15));
    btnList.setSize(100, 20);
    btnList.setLocation(340, 250);
    btnList.addActionListener((ActionEvent e) -> {
        setVisible(false);
        ListagemView li = new ListagemView(controller.toList());
    });
    c.add(btnList);

    setVisible(true);
}

}

public class ListagemView extends JFrame {

    final protected Container c;
    final private JLabel typeLabel;
    final private JTextArea listTextArea;

    public ListagemView(String textArea) {
        setBounds(300, 90, 600, 800);
    }
}

```

```

setDefaultCloseOperation(EXIT_ON_CLOSE);
setResizable(false);

c = getContentPane();
c.setLayout(null);

typeLabel = new JLabel("Listagem");
typeLabel.setFont(new Font("Arial", Font.PLAIN, 30));
typeLabel.setSize(150, 100);
typeLabel.setLocation(225, 10);
c.add(typeLabel);

listTextArea = new JTextArea();
listTextArea.setFont(new Font("Arial", Font.PLAIN, 15));
listTextArea.setSize(400, 500);
listTextArea.setLocation(100, 200);
listTextArea.setLineWrap(true);
listTextArea.setText(textArea);
listTextArea.setEditable(false);
c.add(listTextArea);

setVisible(true);
}

}

```

c. Controller

```

public interface IController {

    public void include(String title, int year, String author);

    public String toList();

    public boolean isFieldsNotNull(String title, int year, String
author);

}

public class LivrosController implements IController {

```

```

public Listagem list = new Listagem();

@Override
public void include(String title, int year, String author) {

    if (!isFieldsNotNull(title, year, author)) {
        return;
    }

    Livros l = new Livros(title, year, author);

    System.out.println(l.print());

    list.library.add(l);

}

@Override
public String toList() {

    if (list == null) {
        return "";
    }

    String textArea = "";

    for (BaseType library : list.library) {
        if (!textArea.concat(library.print()).isEmpty()) {
            textArea = textArea.concat(library.print());
        } else {
            break;
        }
    }

    System.out.println(textArea);
    return textArea;
}

@Override
public boolean isFieldsNotNull(String title, int year, String
author) {

```

```

        try {
            if (title.isEmpty() || year == 0 || author.isEmpty()) {
                throw new Exception("Alguns campos estão
nulos!!");
            } else {
                return true;
            }
        } catch (Exception e) {
            System.out.println(e.getMessage());
            return false;
        }
    }
}

```

```

public class RevistaController implements IController {

    public Listagem list = new Listagem();

    @Override
    public void include(String title, int year, String org) {
    }

    public void include(String org, int volume, int number,
String title, int year) {

        Revista l = new Revista(org, volume, number, title, year);

        System.out.println(l.print());

        list.library.add(l);
    }

    @Override
    public String toList() {

        if (list == null) {
            return "";
        }
    }
}

```

```

String textArea = "";

for (BaseType library : list.library) {
    if (!textArea.concat(library.print()).isEmpty()) {
        textArea = textArea.concat(library.print());
    } else {
        break;
    }
}

System.out.println(textArea);
return textArea;
}

public boolean isFieldsNotNull(String org, int volume, int
number, String title, int year) {
    try {
        if (title.isEmpty() || year == 0 || number == 0 ||
volume == 0 || org.isEmpty()) {
            throw new Exception("Alguns campos estão
nulos!!");
        } else {
            return true;
        }
    } catch (Exception e) {
        System.out.println(e.getMessage());
        return false;
    }
}

@Override
public boolean isFieldsNotNull(String title, int year, String
author) {
    throw new UnsupportedOperationException("Not supported
yet.");
}
}

```

2. Imagens

Revista

Titulo

Org.

Ano Nro. Vol.

Listagem

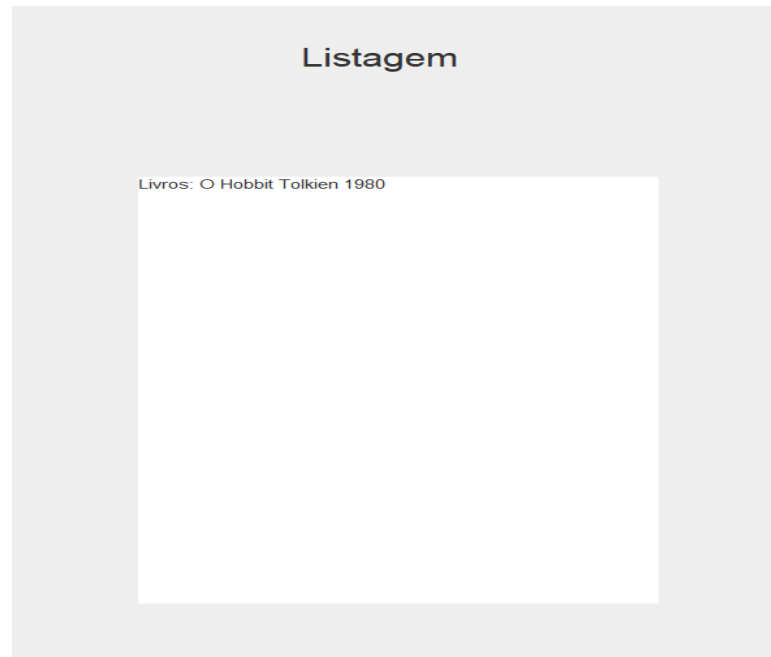
Revista: The 100 CNN 2 120 2010

Livros

Titulo

Autor

Ano



3. Explicação

Os bad smell que percebi nesse código foram o Duplicated Code, na parte dos view, do qual eu não consegui fazer uma generalização boa, resultando em duas classes que contêm código literalmente, além disso alguns métodos estão com overload com muitos parâmetros caracterizando o bad smell Long Parameter List, também possuem classes alternativas e a abstração de certas classes não ficaram tão boas ao ponto de pegar um propósito comum entre elas. Por fim, o código resultante tem alguns Middle Man que poderiam ser removidos porém eu tentei e não conseguir remover por está causa ocorrerá um alto acoplamento do código e a refatoração para a criação da Classe “Jornal” se tornará difícil principalmente nos módulos de View.

Além disso, tentei aplicar os princípios do SOLID no projeto tentando evitar classes longa, o baixo acoplamento, muitas responsabilidades para um classe só e a generalização por meio de classes abstratas para conseguir ter um polimorfismo no código.

