

Linguagem de Programação II

(Programação Orientada a Objetos)

Linguagem de Programação II

<https://github.com/fsantanna-uerj/LP2/>

Francisco Sant'Anna

francisco@ime.uerj.br



Não é um curso de Java!

Formato do Curso

- Google Classroom
- Google Meet
 - Segunda e Quarta M3M4
- Aulas pré-gravadas
 - ? Módulos, ? vídeos, ~??h
- Avaliação
 - **Perguntas e Projeto de Curso**
 - Provas (???)
 - 75% de presença

Google Classroom

- Mural
 - toda comunicação com a turma
- **Atividades**
 - **perguntas/projeto, vídeos e slides, material complementar**
- Notas
 - em cima das perguntas/projeto

Google Meet

- Aulas síncronas
 - Segunda e Quarta M3M4
- Meet
 - <https://meet.google.com/hcj-jxai-isx>
- Conteúdo
 - discussão sobre os vídeos
 - dúvidas sobre as perguntas e o projeto

Aulas Pré-Gravadas

- ? Módulos, ? vídeos, ~??h

1. Introdução, 3 vídeos, ~1h

2. Classes, 6 vídeos, ~2h30

3. ?

4. ?

Avaliação

- Perguntas devem ser respondidas no prazo
 - notas decrescentes conforme atrasos
- **Notas são comparativas**
- ? médias, uma para cada conjunto de perguntas
 - Módulo ? (x?), Módulo ? (x?), Módulo ? (x?), Módulo ? (x?)
- Prova final: ajustes no projeto
- 75% de presença nas aulas síncronas

(Assumindo que não haverá provas presenciais)

Linguagem de Programação II

(Programação Orientada a Objetos)

Linguagem de Programação II

<https://github.com/fsantanna-uerj/LP2/>

Francisco Sant'Anna

francisco@ime.uerj.br



Projeto: Editor Gráfico Vetorial

Parte (1/2)

Linguagem de Programação II

<https://github.com/fsantanna-uerj/LP2/>

Francisco Sant'Anna

francisco@ime.uerj.br

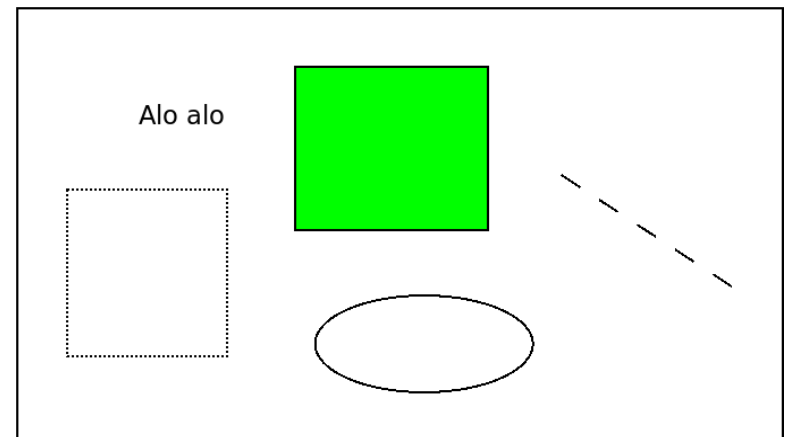


IMPORTANTE

- Use um diretório único para o projeto inteiro
- Não crie novas pastas
- Altere os arquivos já existentes
- Não crie cópias desnecessárias
- Isso é importante para manter o histórico
- Isso é importante para corrigir o trabalho

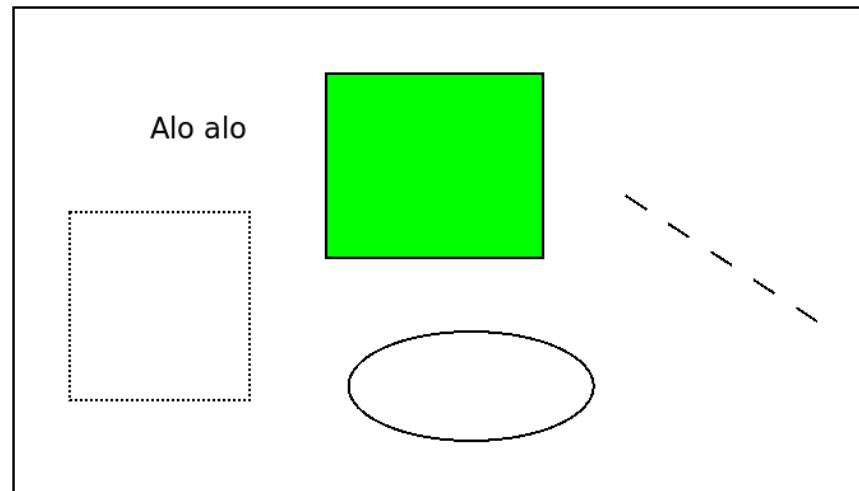
Projeto 1/2

- Pelo menos 4 figuras diferentes
- Criação e remoção
- Propriedades ajustáveis
 - Posição, tamanho e *z-order*
 - Cor de contorno e cor de fundo
- Como?
 - Lista de figuras ativas
 - Listeners de mouse e teclado
 - Figura em foco



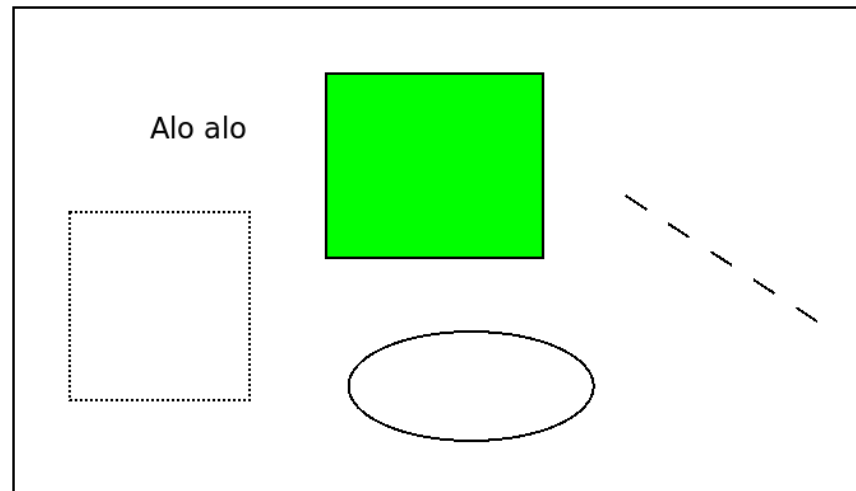
Projeto 1/2

- Pelo menos 4 figuras diferentes
 - Rect, Ellipse, ?, ?
 - Superclasse `Figure` com comportamento comum
 - `x,y,cor, paint, drag ; w,h?`



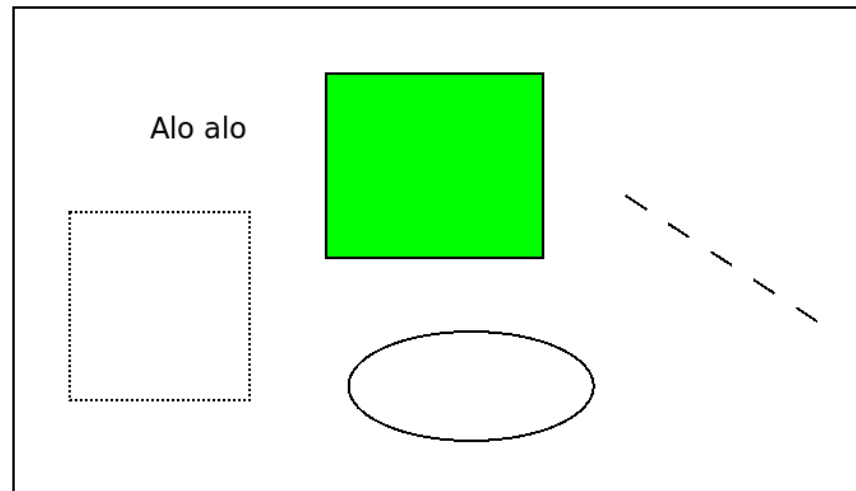
Projeto 1/2

- Criação e remoção
 - Teclas de criação: ‘*r*’, ‘*e*’, ...
 - Tecla *delete* remove figura com foco



Projeto 1/2

- Propriedades ajustáveis
 - Posição, tamanho e *z-order*
 - Cor de contorno e cor de fundo



Projeto 1/2

- Como?
 - Lista de figuras ativas

```
// construtor
ArrayList<Figure> figs = ...

// teclado
figs.add(new Rect(x1,y1, w1,h1));
figs.add(new Ellipse(x2,y2, w2,h2));

// paint
for (Figure fig: figs) {
    fig.paint(g);
}
```

Projeto 1/2

- Como?
 - Listeners de mouse e teclado

```
Frame () {  
    this.addMouseListener(new MouseAdapter() {  
        public void mousePressed (MouseEvent evt) { ... }  
    });  
  
    this.addMouseMotionListener(new MouseMotionAdapter() {  
        public void mouseDragged(MouseEvent evt) { ... }  
    });  
  
    this.addKeyListener(new KeyAdapter() {  
        public void keyPressed (KeyEvent evt) { ... }  
    });  
}
```


Projeto 1/2

- Como?
 - Figura em foco

```
class Frame extends JFrame {  
    ArrayList<Figure> figs = new ...;  
    Figure focus = null;  
    ...  
}
```

```
public void mousePressed (MouseEvent evt) {  
    focus = null;  
    for (Figure fig: figs) {  
        if (fig.x vs evt.getX() && fig.y vs evt.getY()) {  
            focus = fig;  
        }  
    }  
}
```

Dúvidas

- Como pegar a posição atual do mouse?
- Como evitar que as figuras pisquem (flickering)?

Projeto: Editor Gráfico Vetorial

Parte (1/2)

Linguagem de Programação II

<https://github.com/fsantanna-uerj/LP2/>

Francisco Sant'Anna

francisco@ime.uerj.br



Projeto: Editor Gráfico Vetorial

Parte (2/2)

Linguagem de Programação II

<https://github.com/fsantanna-uerj/LP2/>

Francisco Sant'Anna

francisco@ime.uerj.br

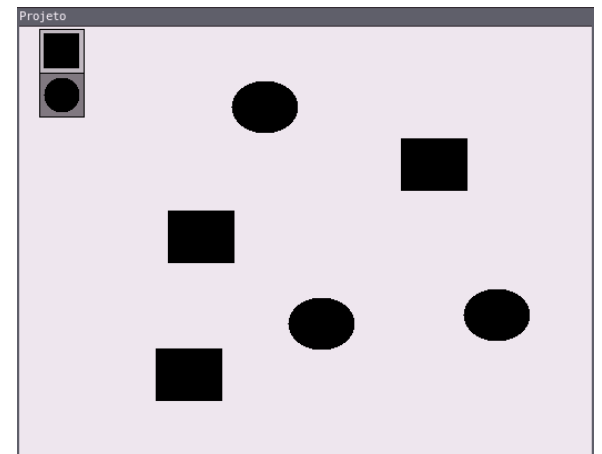


IMPORTANTE

- Use o mesmo repositório do Projeto 1/2
- Use o mesmo diretório do Projeto 2/2
- Não crie novas pastas
- Altere os arquivos existentes
- Não crie cópias desnecessárias
- Isso é importante para manter o histórico
- Isso é importante para corrigir o trabalho

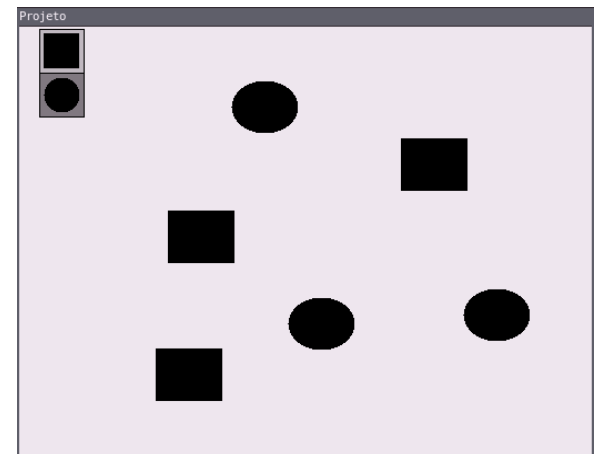
Projeto 2/2

- Menu de botões (*toolbox*)
 - Criar figuras (alternativa ao teclado)
- Gravação em arquivo
 - Arquivo único (*proj.bin*)
 - Gravar ao sair, carregar ao entrar
- Extras
 - Método *clicked* por subclasse
 - Menu mais elaborado
 - Gravar no formato *SVG*



Projeto 2/2

- Menu de botões
- Gravação em arquivo
- Como?
 - IVisible (*clicked*, *paint*)
 - Serializable



Projeto 2/2

- IVisible (*clicked, paint*)

```
// invisible/IVisible.java
package invisible;
public interface IVisible {
    public boolean clicked (int x, int y);
    public void    paint    (Graphics g, boolean focused);
}
```

```
// Button.java
import invisible.IVisible;
public class Button implements IVisible {
    public boolean clicked (int x, int y) { ... }
    public void paint (Graphics g, boolean focused) { ... }
}
```


Projeto 2/2

- IVisible (*clicked, paint*)

```
// figures/Figure.java
package figures;
import invisible.IVisible;
public abstract class Figure implements IVisible {
    public boolean clicked (int x, int y) { ... }
}
```

```
// figures/Rect.java
package figures;
public class Rect extends Figure {
    public void paint (Graphics g, boolean focused) { ... }
}
```

Projeto 2/2

- IVisible (*clicked, paint*)

```
// Button.java
public class Button implements IVisible {
    public int    idx;
    private Figure fig;
    public Button (int idx, Figure fig) { ... }
    public boolean clicked (int x, int y) { ... }
    public void paint (Graphics g, boolean focused) {
        if (focused) { ... } else { ... }
        g2d.drawRect(...);
        this.fig.paint(g, false);
    }
}
```

Projeto 2/2

```
// App.java
class Frame extends JFrame {
    ArrayList<Figure> figs = new ArrayList<Figure>();
    ArrayList<Button> buts = new ArrayList<Button>();
    Button focus_but = null;
    Frame () {
        buts.add(new Button(0, new Rect(0,0,0,0)));
        ...
        public void mousePressed (MouseEvent evt) {
            for (Button but: buts) {
                if (but.clicked(...)) { ... }
            }
        }
    }
    public void paint (Graphics g) {
        ...
        for (Button but: this.buts) {
            but.paint(g, but==focus_but);
        }
    }
}
```

Projeto 2/2

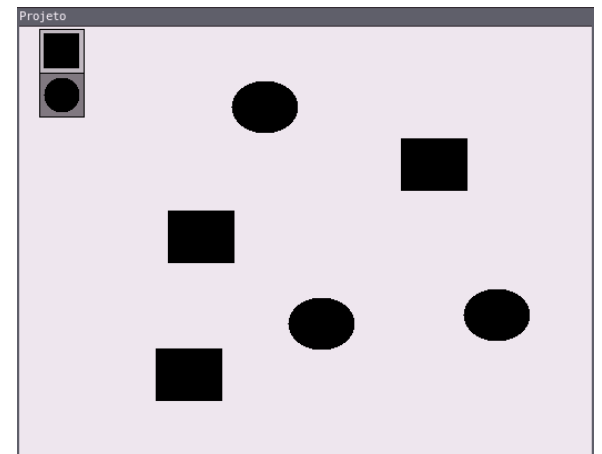
- Serializable

```
// figures/Figure.java  
package figures;  
import invisible.IVisible;  
import java.io.Serializable;  
public abstract class Figure implements IVisible, Serializable {  
    ...  
}
```

```
// App.java
import java.io.*;
class Frame extends JFrame {
    Frame () {
        try {
            FileInputStream f = new FileInputStream("proj.bin");
            ObjectInputStream o = new ObjectInputStream(f);
            this.figs = (ArrayList<Figure>) o.readObject();
            o.close();
        } catch (Exception x) {
            System.out.println("ERRO!");
        }
        ...
        public void windowClosing (WindowEvent e) {
            try {
                FileOutputStream f = new FileOutputStream("proj.bin");
                ObjectOutputStream o = new ObjectOutputStream(f);
                o.writeObject(figs);
                o.flush();
                o.close();
            } catch (Exception x) {
            }
            System.exit(0);
        }
    }
}
```

Projeto 2/2

- Menu de botões (*toolbox*)
 - Criar figuras (alternativa ao teclado)
- Gravação em arquivo
 - Arquivo único (*proj.bin*)
 - Gravar ao sair, carregar ao entrar
- Extras
 - Método *clicked* por subclasse
 - Menu mais elaborado
 - Gravar no formato *SVG*



Projeto: Editor Gráfico Vetorial

Parte (2/2)

Linguagem de Programação II

<https://github.com/fsantanna-uerj/LP2/>

Francisco Sant'Anna

francisco@ime.uerj.br

