

## Aula Prática 3

Docente: Miguel Tavares Coimbra

1. **Extensão de classes.** A filosofia orientada a objectos do Java permite-nos usar a hereditariedade, como uma base de partidas para novas classes que aumentam ou alteram as funcionalidades de classes já existentes.

- a. Crie uma classe que inicialize uma Frame com um botão normal.

```
public class AP4 {  
    public static void main(String[] args) {  
        Button mybutton = new Button();  
        Frame myframe = new Frame();  
        myframe.add(mybutton);  
        myframe.setSize(100,100);  
        myframe.setVisible(true);  
    }  
}
```

- b. Dentro do mesmo projecto, crie uma classe OkButton que estende a classe AWT Button, de forma a que todos estes novos botões tenham algumas características específicas.

```
public class OkButton extends Button {  
    public OkButton () {  
        super("");  
    }  
}
```

- c. Altere o construtor da classe, adicionando-lhe uma linha que garante que o texto visível no botão é “OK”. [Opcional] Altere também a fonte e a cor do botão. Pista: verifique os métodos da classe Component. <http://java.sun.com/j2se/1.4.2/docs/api/java/awt/Component.html>
- d. Crie uma classe Frame, que use um GridLayout, e adicione-lhe 4 componentes: Dois MyButton e dois Button.

2. **Interfaces.** Uma funcionalidade importante do Java é a possibilidade de implementar interfaces específicos em certas classes. Uma utilidade fundamental deste mecanismo é possibilitar que os vários componentes AWT possam ser consumidores de eventos.

- a. Crie uma classe do tipo Frame, que implemente a interface de ActionListener. Para tal, garanta que a classe possui a função actionPerformed.

```
class MyFrame extends Frame implements ActionListener { }
```

- b. Implemente uma funcionalidade específica para quando a classe receber um evento do tipo Action: Altere o título da janela, alternando entre os textos “Janela Fixe” e “Janela Foleira”.

- c. Crie um botão associado à classe MyFrame, registando esta última como ouvinte deste.  
*mybutton.SetActionListener(this);*
  - d. Confirme que ao carregar no botão, a janela muda efectivamente de título.
3. **Eventos.** Usando extensões e interfaces, podemos alterar um TextField tradicional para ter as funcionalidades que precisamos para a nossa aplicação de Sudoku.
- a. Crie uma classe que estenda um TextField e que implemente a interface de KeyListener.  
<http://java.sun.com/j2se/1.4.2/docs/api/java/awt/event/KeyListener.html>
  - b. Altere a função keyTyped de modo a que o seu TextField ignore as teclas pressionadas que não sejam números ou backspace.  
<http://java.sun.com/j2se/1.5.0/docs/api/java/awt/TextField.html>
  - c. Altere a função keyTyped de tal forma que garanta que o seu TextField apenas possa conter um único algarismo. Decida se, perante uma nova tecla premida, o algarismo anterior deverá ser mantido ou substituído pelo novo.