

Técnico em Desenvolvimento de Sistemas

Lógica de Programação II

Interface Gráfica Layouts

Alex Helder Cordeiro do Rosário de Oliveira

Instituto Federal de Brasília - *Campus* Brasília

2º semestre de 2018

Motivação

- Lembrando:
 - Fazemos a herança;
 - Crio o objeto da janela no `main()`;
 - Defino um tamanho para a janela;
 - Faço a janela aparecer na tela.
 - Crio todos os componentes de minha janela como atributos dela;
 - Instancio os componentes;
 - Os acrescento à janela com o comando `add()`;
- Mas só aparece o último inserido...

Inserção de Componentes - Layout

- Para se inserir diversos componentes sem problemas, deve ser selecionado um layout;
- Os Layouts são formas de se organizar os componentes dentro de um “contêiner”*.
- Para definir um layout, devemos usar o método `setLayout()`, passando como argumento um objeto que seja `Layout`*.
- Ex:

```
setLayout(new FlowLayout());
```

*Contêiner são componentes de interface gráfica onde podem ser inseridos outros componentes de interface gráfica.


*Implementa a interface `LayoutManager`.

Layouts

- Existem diversos tipos de layouts, cada um indicado para um objetivo.
 - `FlowLayout`;
 - `GridLayout`;
 - `GridBagLayout`;
 - `BorderLayout`;
 - `CardLayout`;
 - `BoxLayout`;
 - `GroupLayout`;
 - ...
- “`NullLayout`”.


FlowLayout

- Os componentes ficam um ao lado do outro;
- Se a largura dos componentes juntos ultrapassar a largura da janela, é feita uma “quebra de linha”.
- Os elementos são apresentados na ordem em que é feito o `add()` no código fonte;
- Preenchidos da esquerda para a direita e depois descendo para a próxima “linha”.
- Não existe muito alinhamento neste layout.

*Exemplo encontrado no arquivo `ExemploFlowLayout.java`. 

GridLayout


- Os componentes são organizados em um formato de tabela com linhas e colunas;
- Todas as linhas tem mesma altura e todas as colunas tem mesma largura;
- Posição dos componentes depende da ordem de inserção (execução do `add()`);

*Exemplo encontrado no arquivo `ExemploGridLayout.java`. 

GridBagLayout

- Os componentes são organizados em um formato de tabela assim como o GridLayout;
- Mas permite a utilização de “células mescladas”;
- Para isso, usa objetos chamados de *constraints**;
- O layout vincula cada componente com um constrain.

*Traduzindo do inglês: Restrições; São objetos da classe GridBagConstraints.

†Exemplo encontrado no arquivo `ExemploGridBagLayout.java`. 


BorderLayout

- Os componentes são organizados de forma a preencher a direção onde foram indicados.
 - NORTH
 - SOUTH
 - EAST
 - WEST
 - CENTER
- Direções que não foram usadas serão preenchidas pelo componente vinculado ao centro.
- A indicação da direção ocorre como um segundo argumento do `.add()`.

*Exemplo encontrado no arquivo `ExemploBorderLayout.java`.


CardLayout

- O layout utiliza diversos “*cards*” diferentes;
- Cada *card* é um contêiner onde se podem inserir diversos componentes e organizá-los de acordo com um layout próprio;
- Inserimos diversos *cards* no layout e temos apenas um visível por vez.
- A mudança de *card* deve ser prevista em código.

*Exemplo encontrado no arquivo `ExemploCardLayout.java`. 

BoxLayout

- Organiza os componentes de forma semelhante ao `FlowLayout`;
- Permite maior especificação de como se deseja organizar os componentes;
- Permite definir se os componentes devem ser inseridos na horizontal ou na vertical;

*Exemplo encontrado no arquivo `ExemploBoxLayout.java`. 

GroupLayout


- Cria grupos verticais e horizontais;
- Os grupos permitem organizar os componentes de forma alinhada entre os grupos.

*Exemplo encontrado no arquivo `ExemploGroupLayout.java`

“NullLayout”

- Não é um layout de verdade*.
- Usamos `setLayout(null)` para deixar a janela sem layout;
- Inserimos os objetos;
- Usamos o `setBounds()` para definir a posição ocupada pelo componente.
- 4 Argumentos:
 - posição inicial horizontal, posição inicial vertical (canto superior esquerdo);
 - largura, e altura.
- Muito trabalhoso para calcular onde pretende-se colocar cada componente;
- Problemas relacionados a redimensionamento da janela;
- Falta de relacionamento com barras de rolagem (`JScrollBar`).

*Não existe objeto implementando `LayoutManager`.

*Exemplo encontrado no arquivo `ExemploNullLayout.java`. 

Contêineres

- Os objetos de Swing ou de AWT que podem ser inseridos em uma janela gráfica são chamados de Componentes (herdam de Component).
- Existem alguns Componentes que permitem a inserção de outros componentes.
- Estes componentes são chamados de Contêineres (herdam de Container, que por sua vez herda de Component).

Contêineres

- Os objetos de Swing ou de AWT que podem ser inseridos em uma janela gráfica são chamados de Componentes (herdam de Component).
- Existem alguns Componentes que permitem a inserção de outros componentes.
- Estes componentes são chamados de Contêineres (herdam de Container, que por sua vez herda de Component).
- Em um contêiner, podemos inserir qualquer tipo de componente;
- Inclusive outros contêineres;

Contêineres

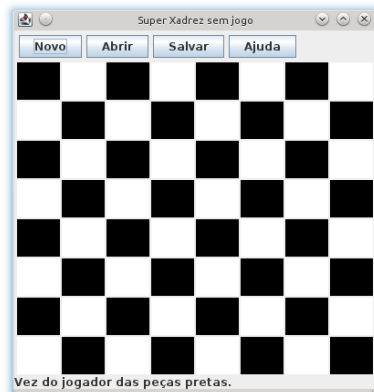
- Cada layout é associado a um contêiner;
- Diferentes contêineres podem ter diferentes layouts, mesmo que estejam na mesma janela;
- Assim, uma janela pode ter diversos layouts; um em cada contêiner.

Contêineres

- Cada layout é associado a um contêiner;
 - Diferentes contêineres podem ter diferentes layouts, mesmo que estejam na mesma janela;
 - Assim, uma janela pode ter diversos layouts; um em cada contêiner.
-
- Janelas agradáveis à vista e intuitivas, muitas vezes, dependem de uma boa combinação de diferentes layouts;
 - A boa escolha dos layouts é tarefa do programador;
 - Esta escolha é um dos fatores que pode levar o programa a ser um sucesso (ou um desastre).

Contêineres

- A figura abaixo ilustra a composição com diversos layouts:
- A janela usou o BorderLayout para poder criar a barra de ferramentas (no NORTH) e a barra de status (no SOUTH); O tabuleiro ocupou o CENTER;
- A barra de ferramentas foi criada com um JPanel com o FlowLayout, pois precisávamos apenas que os botões ficassem um atrás do outro;
- O tabuleiro foi feito com um JPanel usando um GridLayout de 8 linhas e 8 colunas.



*Exemplo encontrado no arquivo Tabuleiro.java.