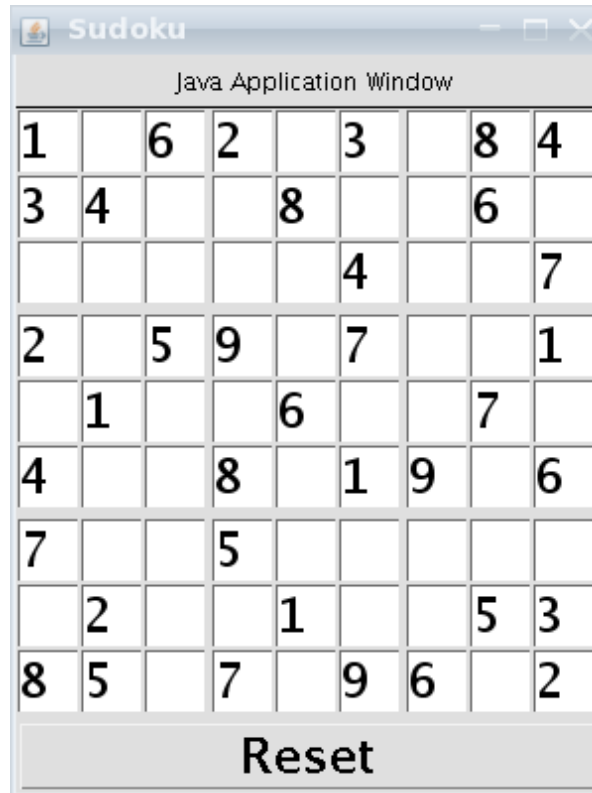


## Aula Prática 4

Docente: Miguel Tavares Coimbra

**Criação de protótipos em Java.** O objectivo será implementar um protótipo de um software que permita jogar Sudoku em várias plataformas computacionais (desktop, laptop, PDA).



### 1. Criação de uma interface gráfica adaptável à resolução do ecrã.

- a. Crie um esquema com a relação hierárquica entre os vários componentes visuais AWT a utilizar. Use quadrados para os componentes e setas para a ligação entre componentes e contentores. Anote ao lado dos contentores que *Layout Manager* planeia utilizar.
  - i. Use um contentor do tipo *Frame* como componente topo da hierarquia. Se escolher um *BorderLayout* poderá colocar a *Sul* o botão (*Button*) de 'Reset', e ao centro um painel (*Panel*).
  - ii. Use um *Grid Layout* para o painel anterior, de tamanho 3x3, criando 9 novos painéis, um para cada posição. Garante que o espaçamento entre elementos da grelha é igual a 2, de forma a distinguir visualmente as casas do Sudoku.
  - iii. Dentro de cada um destes novos painéis, coloque 9 componentes do tipo *TextField*, organizando-os segundo um *Grid Layout* de tamanho 3x3. Guarde referências a cada um destes *TextField*, usando para o efeito um vector de tamanho 81, previamente inicializado.

- b. Implemente em Java o esquema que desenhou.
  - i. Crie um novo projecto.
  - ii. Comece por instanciar e formatar todos os componentes e contentores.
  - iii. Adicione *Layout Managers* a todos os contentores.
  - iv. Adicione os componentes adequados a cada contentor.
  - v. Torne visível a janela.

## 2. Implementação da funcionalidade ‘Clear’.

- a. Garanta que a classe principal do projecto estende ela mesmo o componente *Frame*.
  - i. `class AP4 extends Frame`
  - ii. Não se esqueça de fazer o instanciamento dinâmica da classe principal (dando uma ordem de `new AP4()` dentro da função `main`, e deslocando o código de inicialização para dentro da função construtora `public AP4()`);
- b. Após assegurar que a interface continua a ser correctamente criada, garanta que a sua classe implementa a interface de *ActionListener*.
  - i. `class AP4 extends Frame implements ActionListener`
  - ii. A classe deverá implementar a função `actionPerformed`.
- c. Adicione a classe *AP4* como consumidora dos eventos produzidos pelo botão de ‘Clear’.
  - i. `button.addActionListener(this);`
- d. Dentro da função `actionPerformed`, crie um ciclo que se encarregue de colocar o texto de cada um dos 81 *TextField* igual a “”.

## 3. Implementação das funcionalidades associadas ao teclado.

- a. Tal como fez com os *ActionEvents*, garanta que a sua classe principal tem capacidade de consumir eventos de teclado (*KeyEvents*).
  - i. A classe deverá implementar a interface de *KeyListener*.
- b. Altere a função `keyTyped` de forma a controlar que apenas um carácter é inserido em cada *TextField*, assim como que este apenas deverá ser um número de 1 a 9. Deverá também lidar com a tecla de *backspace*.

## 4. [Avançado] Implemente as funcionalidades do jogo Sudoku

- a. Trabalhe o vector de 81 *TextField*, sempre que um novo carácter é teclado, de forma a impedir que sejam inseridos números que violam as regras do Sudoku.
- b. Assinale a vermelho (alterando a cor de fundo do componente) os *TextField* que foram responsáveis pelas violações detectadas nos passos anteriores.