

## Lab III.

### Objetivos

Os objetivos deste trabalho são:

- Aplicar conceitos de modulação de software necessários no desenvolvimento de uma solução
- Rever e consolidar competências de desenvolvimento de software

### III.1 Jogo do Galo

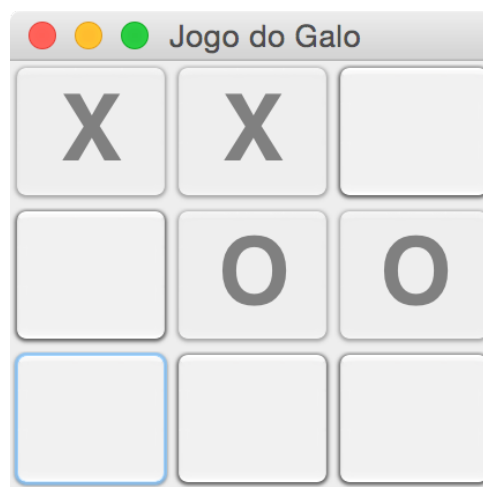
Pretende-se desenvolver uma versão simples do “Jogo do Galo”. Para tal são fornecidos os seguintes módulos (no dossier das aulas práticas):

- a) A aplicação visual do jogo, classe *JGalo*, desenvolvida sobre java Swing e que não precisa de ser modificada para este trabalho. Apesar disso, recomenda-se a sua análise cuidada.
- b) A interface *JGaloInterface* que irá servir de ligação entre a classe *JGalo* e o módulo que terá de desenvolver.

```
public interface JGaloInterface {  
    public abstract char getActualPlayer();  
    public abstract boolean setJogada(int lin, int col);  
    public abstract boolean isFinished();  
    public abstract char checkResult();  
}
```

Considere que o programa apenas executa o jogo uma vez, começando com cruzeiros (X) ou bolas (O) consoante o argumento inicial (por omissão, considere X).

Nota: No dossier existe ainda um ficheiro executável (*JogoDoGalo.jar*) que permite verificar o comportamento desejado para este programa.



## III.2 Rua

O objetivo é desenvolver um programa para gerir a localização de famílias e seus membros ao longo de uma rua. Analise os requisitos e planeie cuidadosamente as interfaces, classes, e estruturas de dados mais adequadas.

### Requisitos iniciais

1. A numeração de porta começa em 1.
2. Cada membro pode ocupar um ou mais números  $[x1, x2]$ , em que  $0 < x1 \leq x2$ .
3. Dois membros com o mesmo intervalo de localização são considerados membros da mesma família.
4. Cada família tem um ou mais membros nomeados da seguinte forma:
  - a. O nome é único
  - b. Um nome contém apenas letras (maiúsculas e minúsculas), dígitos numéricos e estes símbolos: `_.`@
  - c. O nome deve começar com uma letra.
  - d. As letras podem ser maiúsculas ou minúsculas.
  - e. O nome não pode acabar com um símbolo.
  - f. O comprimento máximo de um nome é de 30 caracteres.
  - g. O nome deve ter pelo menos 3 caracteres.

### Comandos

O programa deve permitir os seguintes comandos, lidos da consola (*case insensitive*):

Load, Map, Add, Remove, List, Lookup, Clear, Quit  
Command:

- **Load *filename*:** Lê um ficheiro de texto para construir uma representação interna da rua. Este arquivo pode ter uma linha de descrição inicial que começa com um caracter ">". As linhas seguintes contêm um membro por linha na forma "*Número Inicial-NúmeroFinal Nome*". Esta lista pode aparecer em qualquer ordem.

Exemplo de ficheiro:

```
> Exemplo de teste
10-12 nola
11-14 mikey@ua.pt
4-7 Ana_98
4-7 mikey
10-17 rui
10-17 wx1
16-18 pete
5-6 falB
16-18 silvia
16-18 sergi
```

- **Map:** mostra o mapa da rua no formato seguinte. As famílias devem ser listadas de forma decrescente pela sua abrangência. Além das colunas 'Family' e 'Name' devem ser adicionadas colunas referentes a cada número de porta, as quais são preenchidas com 'X' em caso de ocupação.

Exemplo de Mapa:

FAMILY	NAME	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	rui							X	X	X	X	X	X	X	X	
1	wx1							X	X	X	X	X	X	X	X	
2	<a href="mailto:mikey@ua.pt">mikey@ua.pt</a>								X	X	X	X				
3	Ana_98	X	X	X	X											
3	mikey	X	X	X	X											
4	nola							X	X	X						
5	pete													X	X	X
5	silvia													X	X	X
5	sergi													X	X	X
6	falB		X	X												

- **Add nome x1 x2:** cria um novo membro na localização x1, x2. Se o membro já existir deve ser reportado um erro.
- **Remove name:** remove um membro da família. Se o membro não existir deve ser reportado um erro.

(Os seguintes comandos são opcionais)

- **List:** apresenta todos os nomes e números de porta, ordenados alfabeticamente.

```
aaa 1 2
bbb 1 2
cnd 1 2
sm7_a 5 6
```

- **Lookup name:** procura um habitante e apresenta a informação da família, na forma:  
x1 x2 : name1 name2

- **Clear:** limpa toda a informação da rua.

- **Quit:** termina o programa.

Deve permitir que o programa possa ser executado usando um ficheiro de comandos como argumento de entrada. Por exemplo, "java lab3 ficheiro\_de\_comandos". No dossier da disciplina são fornecidos alguns ficheiros de exemplos de distribuição de habitantes.

**Nota importante:** para cada guião prático, deverá ser usada no *git* uma nomenclatura uniforme (*lab01*, *lab02*, *lab03*,...) para permitir uma identificação mais fácil dos projetos.

*Bom trabalho!*