

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

3.01 Jogo do Galo

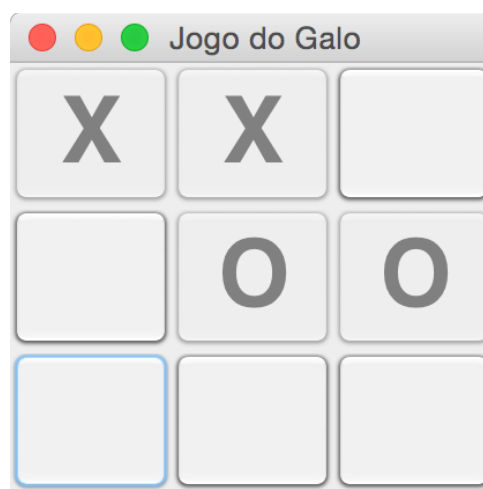
Pretende-se desenvolver uma versão simples do “Jogo do Galo”. Para tal são fornecidos os seguintes módulos (no dossier das aulas práticas):

- 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.



3.02 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 um símbolo.
 - f. O comprimento máximo de um nome é de 40 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 tre
10-17 wx1
16-18 pete
5-6 falB
16-18 silvia
16-18 sergi
```

- **Map:** mostra o mapa da rua no formato seguinte. Habitantes com intervalos sobrepostos são exibidas em colunas paralelas.
Se duas, ou mais famílias, começarem no mesmo valor, a que tem o maior alcance aparece no lado esquerdo. Cada local ocupado por uma família deve ser separado por dois pontos.

Exemplo de Mapa:

```

1
2
3
4 : mikey Ana_98
5 : mikey Ana_98 : falB
6 : mikey Ana_98 : falB
7 : mikey Ana_98
8
9
10 : rui wx1 : nola
11 : rui wx1 : nola : mikey@ua.pt
12 : rui wx1 : nola : mikey@ua.pt
13 : rui wx1      : mikey@ua.pt
14 : rui wx1      : mikey@ua.pt
15 : rui wx1
16 : rui wx1 : pete silvia sergi
17 : rui wx1 : pete silvia sergi
18          : pete silvia sergi

```

- **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.

```

a 1 2
b 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 aula prática, deverá ser usada no *git* uma nomenclatura uniforme (*aula1*, *aula2*, *aula3*, ...) para permitir uma identificação mais fácil dos projetos.

Bom trabalho!