

Unidade Curricular: Programação II

Curso: Engenharia Informática

2º Ano

1º Semestre

Objetivo:

Este trabalho tem como objetivo a aplicação dos conceitos de programação orientada aos objetos através do desenvolvimento de uma aplicação na linguagem de programação Java. A finalidade do presente trabalho é desenvolver uma aplicação que responda aos requisitos enunciados, usando classes com uma estrutura e comportamento adequados, recorrendo à exploração e aplicação dos conteúdos de OO como classes, herança e classes abstratas.

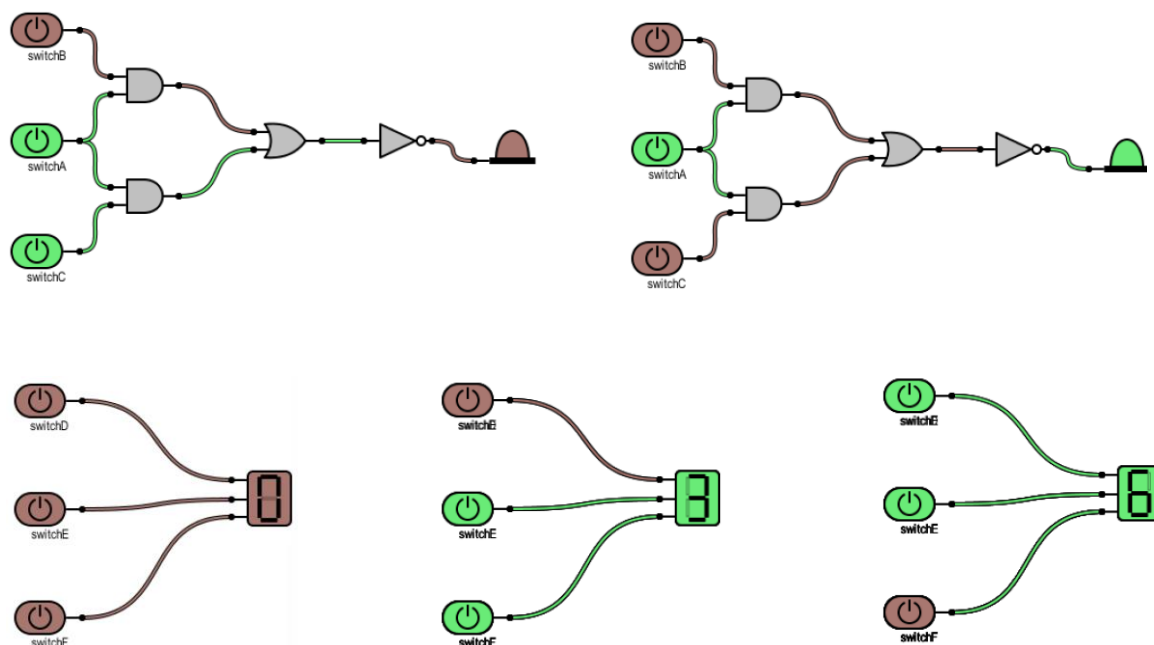
Descrição do trabalho

Tema do problema:

A finalidade do presente trabalho é desenvolver uma aplicação que permita a representação de circuitos lógicos digitais. O programa deve permitir:

- criar circuitos digitais resultantes da combinação de portas lógicas, controlos de entrada (switch) e controlos de saída (led e mostrador de 3 bits);
- realizar o teste do circuito digital através da alteração do estado dos controlos de entrada e visualização automática do resultado nos controlos de saída,

conforme exemplificado nas imagens seguintes:



Requisitos funcionais mínimos:

A aplicação deve ser desenvolvida na linguagem Java, segundo o paradigma de orientação aos objetos, e deve satisfazer os seguintes requisitos funcionais mínimos numa primeira versão:

1. Ter uma interface em modo texto que permita ler comandos.
2. Permitir a visualização esquemática do circuito, efetuada recorrendo à utilização da classe *LCDFrameCmd* fornecida com o enunciado do trabalho prático.
3. Efetuar a definição do *layout* do circuito através da introdução de comandos em formato texto de acordo com as seguintes especificações:

- a. Criar portas lógicas (portas do tipo AND, OR, XOR, NOT) e controlos de entrada (switch) e de saída (led e mostrador de 3 bits) com o comando:

ADD <id>:<tipo_porta>@<coord_X>,<coord_Y> [<legenda>]

em que:

- <id> é um identificador único para cada componente (sem espaços)
- <tipo_porta> pode ser um dos seguintes valores AND, OR, XOR, NOT, LED, 3BD ou SWITCH
- <coord_X>,<coord_Y> são um par de valores inteiros com a posição do elemento na representação gráfica
- <nome_porta> parâmetro facultativo que é a legenda a apresentar na parte inferior do elemento na representação gráfica

- b. Definir as conexões entre elementos do circuito através da indicação do par saída/entrada com o comando:

WIRE <id_elemento1> <id_elemento2> <input_pin_elemento2>

em que:

- <id_elemento1> e <id_elemento2> correspondem aos identificadores definidos nos comandos **ADD**
- <input_pin_elemento2> identifica o pin de ligação da conexão no elemento2, podendo ter os valores PIN_A, PIN_B ou PIN_C.

4. Permitir a alteração do estado de um controlo de entrada com o comando:

TURN <state> <id_elemento_entrada>

em que:

- <state> pode ter o valor ON ou OFF
- <id_elemento_entrada> é o identificador de um dos controlos de entrada (switch) definidos nos comandos **ADD**
- Os elementos são todos criados com o state a OFF

5. Atualizar de forma automática a representação gráfica do circuito, nomeadamente o estado dos controlos de saída e das conexões, com base no estado dos controlos de entrada e das conexões existentes entre os elementos do circuito.
6. Gravar a estrutura de um circuito digital num ficheiro e abrir o mesmo com os comandos:

SAVE <nome_do_ficheiro>

OPEN <nome_do_ficheiro>

A seguir apresenta-se um exemplo do resultado pretendido:

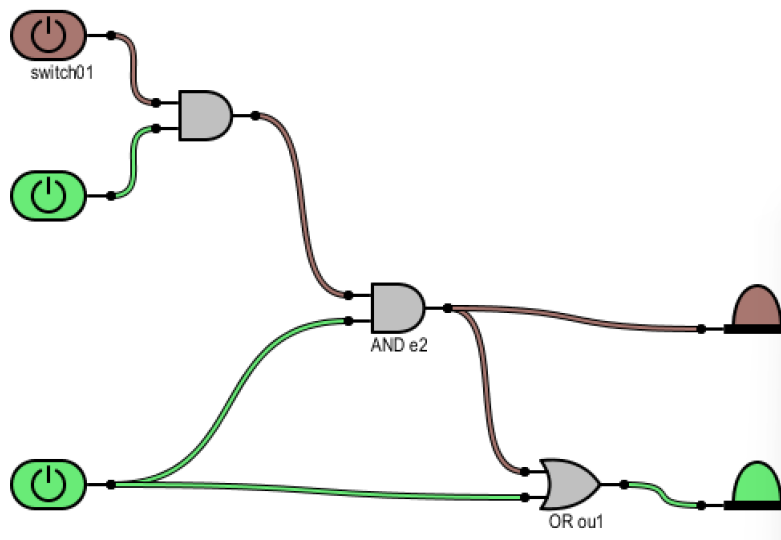
Exemplo: a execução dos seguintes comandos

```

ADD sw1:SWITCH@10,50 switch1
ADD sw2:SWITCH@10,150
ADD sw3:SWITCH@10,330
ADD and1:AND@100,100
ADD and2:AND@220,200 AND e2
ADD or:OR@330,330 OR ou1
ADD led1:LED@440,220
ADD led2:LED@440,330
WIRE sw1 and1 PIN_A
WIRE sw2 and1 PIN_B
WIRE and1 and2 PIN_A
WIRE sw3 and2 PIN_B
WIRE sw3 or PIN_B
WIRE and2 or PIN_A
WIRE and2 led1 PIN_A
WIRE or led2 PIN_A
TURN ON sw2
TURN ON sw3

```

deve resultar no seguinte diagrama:



Funcionalidades adicionais:

Além dos requisitos mínimos enunciados, cada grupo é livre de implementar funcionalidades adicionais para valorizar o trabalho, desde que estas funcionalidades não modifiquem os requisitos obrigatórios e não alterem o objetivo do trabalho.

Exemplos de funcionalidades adicionais que podem ser incluídas na versão do programa com interface em modo texto são:

- Remover um elemento (e conexões associadas)
- Permitir alterar a posição de um dado elemento (e respetivas conexões);
- Listar os elementos que compõe o circuito e respetivas conexões;
- Incluir outras portas lógicas como NAND, NOR;
- Gerar a tabela de verdade de acordo com a estrutura do circuito (uma coluna por controlo de entrada e por controlo de saída em que cada linha corresponde à combinação de valores possíveis de entrada);
- Validar o circuito lógico (ex.: verificação de *loops* lógicos, elementos não conectados, etc.);
- Definir a dimensão da janela gráfica de representação do circuito,
- Apresentação automática sequencial das combinações dos possíveis valores dos controlos de entrada (animação).

No caso de pretenderem implementar funcionalidades mais avançadas, pode ser desenvolvida uma nova versão do programa (a gravar em pasta independente conforme indicado abaixo). Alguns exemplos de funcionalidades extra são:

- Possibilidade de definir portas lógicas com mais de duas entradas;
- Interface gráfica de definição de elementos;
- Funcionalidades de interação com elementos gráficos com a utilização do rato.

NOTA: Independentemente das funcionalidades implementadas, deve ser mantida a representação gráfica dos elementos.

Regras gerais

Requisitos obrigatórios de implementação:

- Além dos requisitos funcionais indicados, a implementação do trabalho prática deve ter em conta os seguintes requisitos obrigatórios de implementação:
 - Definir uma hierarquia de classes (**herança**), recomendando-se a utilização de classes abstratas. **O incumprimento deste requisito tem como consequência a reprovação** na componente de avaliação do trabalho prático.
 - Utilizar coleções para armazenar os elementos dos circuitos (*Map*, *List* ou *Set*);
 - Todos os elementos devem ser serializáveis para permitir o seu armazenamento em ficheiro;
- Juntamente com o enunciado, encontra-se disponível o código fonte de um *package* **logiccircuit** com a definição de classes que deverão ser utilizadas para a implementação do projeto. O código fonte disponibilizado inclui ainda 2 exemplos (*Demo1* e *Demo2*) de utilização das classes do *package* *logiccircuit*, a documentação *JavaDoc* do *package* *logiccircuit* na pasta “JavaDoc - Package logiccircuit”
- A utilização das classes do *package* **logiccircuit** é um requisito obrigatório. As operações gráficas deverão ser efetuadas recorrendo exclusivamente aos métodos das classes *LCDFrameCmd* e *LCDPanel* (sem modificações do seu código fonte).

Orientações para a implementação do trabalho:

- Apenas se pode utilizar a linguagem Java, com base no paradigma de orientação a objetos, para o desenvolvimento do trabalho proposto.
- A classe com o método *main* deve ter **obrigatoriamente o nome ProgCircuito**.
- O trabalho deve ser desenvolvido para funcionar em qualquer plataforma (Linux, Windows, MacOS).

- De forma a cumprir com os requisitos mínimos da aplicação, recomenda-se o desenvolvimento do trabalho prático em duas fases:
 - Numa primeira fase, cada grupo tem de desenvolver uma primeira versão da aplicação que cumpra com os requisitos funcionais mínimos, em particular o funcionamento através da introdução de comandos no terminal em modo texto.
O código fonte das classes desta versão do programa (e outros ficheiros necessários à sua execução) deve ser gravado numa pasta com o nome “src_fase1”.
 - Cumprida a primeira fase, cada grupo pode implementar funcionalidades adicionais no programa desde que não comprometam as funcionalidades desenvolvidas anteriormente (funcionalidades mínimas).
 - No caso de pretenderem desenvolver funcionalidades adicionais que obrigam à alteração significativa do modo de funcionamento do programa, devem conservar a versão do programa anterior na pasta com o nome “src_fase1”, e poderão desenvolver uma nova versão mais avançada do programa, sendo o código fonte gravado numa pasta com o nome “src_fase2”.
- No caso da opção pelo desenvolvimento da segunda fase com funcionalidades mais avançadas, as classes anteriores poderão ser atualizadas no sentido de incluírem outras operações de suporte ao desenvolvimento das novas funcionalidades. Estas alterações deverão ser devidamente documentadas em secção independente do relatório, intitulada “Alterações das Classes de Representação Gráfica”.
- Quaisquer dúvidas de interpretação do presente enunciado devem ser elucidadas junto do docente.
- Pormenores adicionais na implementação de funcionalidades extra são deixados ao critério do grupo de trabalho, devendo estas decisões ser devidamente explicadas na secção independente do relatório, intitulada “Funcionalidades adicionais”.

Entrega e Avaliação

Grupos:

O trabalho prático deve ser concretizado em grupos de dois alunos.

A fim de facilitar a comunicação entre o grupo e o docente, recomenda-se que os alunos de cada grupo estejam inscritos no mesmo turno prático (não obrigatório).

A lista final dos grupos e respetiva numeração será publicada na página da unidade curricular.

Entregas de elementos de avaliação:

A entrega do trabalho prático deve incluir obrigatoriamente:

- Relatório de acordo com o modelo disponibilizado na plataforma Moodle com o máximo de 10 páginas (sem considerar a capa, índice e imagens), no qual todas as funcionalidades devem ser descritas na respetiva secção do relatório, com detalhes sobre a lógica de implementação e a motivação para tal funcionalidade.
- Manual de utilização da aplicação (até 3 páginas), descrevendo a utilização do software sob o ponto de vista do utilizador. A entrega deste manual só é necessária se forem implementadas funcionalidades adicionais relativamente aos requisitos mínimos;
- Código fonte em Java para execução da aplicação (em que a classe com o método *main* deve ter **obrigatoriamente o nome *ProgCircuito***)

A entrega deve ser efetuada através da submissão no Moodle de um ficheiro comprimido (ZIP) com o nome **grupoNN** (em que NN corresponde ao número do grupo identificado com 2 dígitos). O ficheiro comprimido deve incluir os seguintes elementos:

- Ficheiro (DOCX ou PDF) com o relatório;
- Ficheiro (DOCX ou PDF) com o manual de utilização (só no caso de existirem funcionalidades adicionais);
- Pasta com o nome **src_fase1** com os ficheiros Java do programa que cumpre os requisitos funcionais mínimos (isto é, tem de funcionar através de comandos inseridos no terminal);
- A documentação JavaDoc caso tenha sido gerada;
- No caso de terem optado por desenvolver funcionalidades avançadas que não permitem a execução do programa em modo comando, incluir uma pasta com o nome **src_fase2** com os ficheiros Java dessa versão do programa.

AVISO: os trabalhos que **não respeitem estas regras de entrega serão penalizados**, podendo não serem aceites e avaliados.

Datas de entrega:

- **06/01/2025:** Submissão na plataforma Moodle de um único ficheiro comprimido (formato ZIP ou RAR) com o relatório, manual de utilização e a(s) pasta(s) com o código fonte do projeto de acordo com as regras definidas (a submissão do ficheiro deve ser efetuada por um dos alunos do grupo).
- **Até 11/01/2025:** Submissão do trabalho na plataforma Moodle com penalização de acordo com a seguinte grelha:

Dia de entrega	07/01/2025	08/01/2025	09/01/2025	10/01/2025	11/01/2025
Penalização	2%	5%	9%	14%	20%

Avaliação:

- Este trabalho prático faz parte da avaliação da componente prática da unidade curricular de Programação II, com nota mínima de 8,0 valores.
- O trabalho prático é obrigatório em qualquer época de avaliação.
- A implementação dos requisitos mínimos indicados no enunciado é obrigatória para que o trabalho prático tenha avaliação positiva. Quer isto dizer que os alunos deverão, por iniciativa própria, acrescentar funcionalidades adicionais para obter classificações mais elevadas.
- Os grupos deverão **começar por garantir o cumprimento dos requisitos mínimos** e só depois desenvolver funcionalidades adicionais, uma vez que **o desenvolvimento de funcionalidades adicionais só será alvo de valorização se cumpridos os requisitos mínimos enunciados.**
- O trabalho prático será defendido numa sessão própria em horário previamente publicado no Moodle. No caso de incompatibilidade do horário, deverá ser contactado o docente.
- A nota é atribuída **individualmente** aos elementos do grupo, após apresentação, visualização, apreciação e discussão do software desenvolvido e do relatório entregue.
- À hora determinada para a apresentação e discussão oral, o software deve estar pronto a ser executado, e o IDE utilizado no desenvolvimento deve estar aberto e pronto a apresentar o código fonte do trabalho.
- Cada discussão oral durará entre quinze e vinte minutos: cinco minutos iniciais para apresentação de todo o software desenvolvido (todas as capacidades a avaliar devem

ser apresentadas nestes 5 minutos); os restantes dez a quinze minutos serão destinados a perguntas dirigidas a cada elemento do grupo **individualmente**.

- Os alunos que não compareçam à apresentação/defesa final do trabalho prático, exceto em casos devidamente justificados, estão reprovados à unidade curricular de Programação II.
- A avaliação do trabalho prático será efetuada tendo em conta os seguintes critérios:
 - inclusão de todos os elementos indicados na ficheiro submetido no Moodle;
 - cumprimentos dos requisitos funcionais mínimos;
 - demonstração da aplicação a funcionar e capacidade de esclarecer e aprofundar as questões levantadas durante a apresentação e defesa final;
 - qualidade da estrutura de classes criadas, nomeadamente na aplicação dos conceitos fundamentais do paradigma de programação orientada aos objetos;
 - legibilidade e qualidade do código fonte (utilização das convenções do Java em termos de nomes, etc.) e respetiva documentação (valorização se forem utilizados comentários em JavaDoc);
 - cumprimentos dos prazos estabelecidos.
- A apresentação de relatórios e/ou implementações **não originais** (que constituam plágio) ou **iguais entre grupos** tem como consequência a imediata atribuição de **nota zero** no trabalho do(s) grupo(s) e a eventuais processos disciplinares.