Licenciatura de Engenharia Informática (2ºAno)

Ano Letivo 2023/2024

PROGRAMAÇÃO ORIENTADA A OBJETOS

- Meta 2 -

Trabalho Prático

Docente responsável pela unidade curricular:

João António Pereira Almeida Durães

Trabalho realizado por:

Margarida Nunes a2022136170@isec.pt

Vítor Couceiro a2022136345@isec.pt

Índice

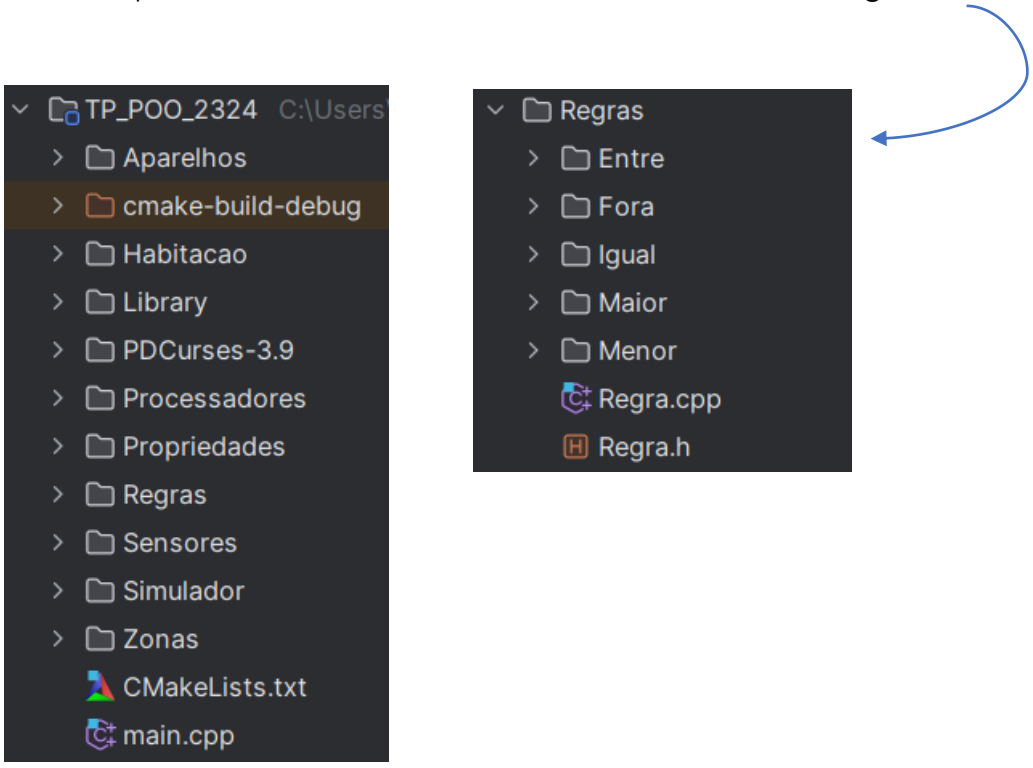
1. Introdução

No âmbito da disciplina de POO foi nos proposto implementar, em C++, um programa que simule alterações de propriedades controladas por vários componentes de domótica, interligados entre si, em várias zonas de uma habitação.

Para a criação do projeto foi necessário fazer uma análise dos conceitos referidos no enunciado do trabalho e organizá-los de modo a conseguir construir classes de objetos coesas e robustas. Ao longo do relatório iremos explicar detalhadamente o funcionamento das classes implementadas no nosso trabalho.

2. Organização do código

Após várias alterações na organização dos ficheiros, esta foi a forma mais prática e organizada que encontrámos. Cada pasta contém o código relativo a cada classe, sendo que nas classes onde foi utilizada herança existem, ainda, uma sub diretoria para cada classe derivada, como é o caso das Regras :



3. Classes implementadas

3.1 Classe Simulação

Responsabilidades:

- Receber e validar o input do jogador;
- Enviar respostas aos jogadores, tendo em conta as ações exercidas por ele, na simulação;
- Única classe que comunica diretamente com o utilizador;
- Faz a chamada da lógica de cada comando do utilizador;
- Responsável por criar e libertar a habitação.

Colaborações: Habitação.

3.2 Classe Habitação

Responsabilidades:

- Guarda uma matriz dinâmica de zonas;
- Elimina / Adiciona zonas;
- Guarda uma coleção das cópias de processadores, realizadas pelo utilizador (com o comando psalva);

Colaborações: Zona

3.3 Classe Zona

Responsabilidades:

- Representar coleções de propriedades, sensores, aparelhos e processadores;
- Gerir essas coleções, adicionando, eliminando ou listando componentes e propriedades em cada zona ;

Colaborações: Propriedade, Sensor, Aparelho, Processador

3.4 Classe Processador

Responsabilidades:

- Tem uma coleção de regras própria.
- Gere essa coleção, adicionando, eliminando ou listando as regras a ele associadas;

Colaborações: Regra, Zona

3.5 Classe Aparelho

Responsabilidades:

- Alterar o valor das propriedades associadas, quando ativado por um processador ou por um comando do utilizador (acom);

Colaborações: Propriedade

3.6 Classe Sensor

Responsabilidades:

- Obter a leitura do valor das propriedades associadas, por indicação do processador.

Colaborações: Propriedade, Processador

3.7 Classe Regra

Responsabilidades:

- Devolve um valor lógico por comparação do valor da propriedade (obtido por um sensor) com o valor de um ou mais parâmetros definidos pelo utilizador, aquando da criação da regra.
- Associa um sensor quando é criada.

Colaborações: Sensor

3.8 Classe Propriedade

Responsabilidades:

- Unidade mais básica e simples de todos os objetos.
- Não tem propriamente uma responsabilidade, apenas certifica-se se o valor que o aparelho tenta colocar está dentro dos seus limites (valor mínimo, valor máximo que pode atingir).

Colaborações: --

4. Criação/destruição dos objetos

- Habitação: a simulação é a classe responsável por criar um objeto habitação e destruí-lo.
- Zona: os objetos desta classe serão criados, armazenados e destruídos na classe Habitação.
- Sensor, Aparelho, Processador e Propriedade: os objetos destas classes serão criados e destruídos na classe Zona.
- Regra: os objetos desta classe serão criados e destruídos na classe Processador.

5. Interface Simulador

No sentido de simplificar e melhorar a interação do utilizador com a simulação apresentada, foi utilizada a biblioteca pdcurses.h (para Windows). A interface foi desenhada para um terminal de dimensões 156x40 (tamanho maximizado), pelo que ligeiras alterações a este formato podem provocar desformatações e/ou comportamentos inesperados e que não podem ser previstos. Em termos de layout, a interface está organizada por janelas:

- Janela de input de comandos.
- Janela de ouput de comandos e informações especificas da simulação.
- (nlinhas x ncolunas) matriz de janelas que representam a habitação – um espaço alocado para a criação de futuras zonas – cujo tamanho é especificado pelo utilizador com o comando 'hnova <nlinhas> <ncolunas>'.

6. Funcionalidades implementadas – requisitos

Componente do trabalho	Realizado
Implementação de todos os comandos do utilizador.	✓
Leitura do ficheiro de comandos (comando exe).	✓
Criação de todas as classes de objetos e suas respetivas funcionalidades.	✓
Realização da cópia em memória de processadores, e respetiva reposição do seu estado (comandos psalva e prepõe).	✓
Funcionalidades prox e avança instantes.	✓
Permitir a adição de novos componentes que não sejam reconhecidos, neste momento, pelo simulador.	parcialmente
Organizar devidamente o projeto em ficheiros .h e .cpp.	✓