

Programação Orientada a Objetos (POO) - LEIM

Trabalho Prático

Luís G. M. Ferreira *

EST-IPCA
Barcelos

13 de outubro de 2023

Resumo

Este trabalho prático da disciplina de Programação Orientada a Objetos (POO) foca a análise de problemas reais simples e a aplicação do Paradigma Orientado a Objetos na implementação de possíveis soluções.

* EMAIL: lufer@ipca.pt; GAB: 5; IPCA-EST

1 Motivação

Pretende-se que sejam desenvolvidas soluções em C# para problemas reais de complexidade moderada. Serão identificadas as principais entidades (classes) envolvidas, as estruturas para suportar os dados e implementados os principais processos capazes de suportar soluções para esses problemas.

Pretende-se ainda contribuir para a boa redação de relatórios que descrevam o trabalho desenvolvido, a boa documentação de código fonte com a geração da API, e a gestão e planeamento de trabalho via ferramentas apropriadas (Git, GitHub, Trello, ou outras).

2 Objectivos gerais

- Consolidar conceitos basilares do Paradigma Orientado a Objectos;
- Analisar problemas reais;
- Desenvolver capacidades de programação em C#;
- Potenciar a experiência no desenvolvimento de software;
- Assimilar o conteúdo da Unidade Curricular.

3 Objectivos Concretos

- Saber identificar e implementar classes e objetos num problema real;
- Saber produzir código com qualidade, bem documentado, de acordo com a norma CLS;
- Saber gerar a API com a documentação do código que produziu (Doxigen ou outro);
- Saber aplicar os pilares da POO: Herança, Encapsulamento, Abstração e Polimorfismo;
- Saber estruturar devidamente uma solução em bibliotecas de classes;
- Saber estruturar uma solução por camadas, seguindo padrões como NTier ou MVC;
- Conseguir definir uma estratégia de persistência de dados;
- Saber analisar o código que produziu;
- Saber executar testes simples sobre o código produzido;
- Saber explorar das vantagens que o C# oferece.

4 Outros Objetivos

- Saber utilizar ferramentas de gestão de versões de código (Git/Github)
- Saber utilizar ferramentas de gestão de projetos.

5 Regras do "Jogo"

- O trabalho deverá ser feito individualmente ou em grupo de duas pessoas. No caso de ser realizado em grupo, deverão previamente comunicar ao docente;
- Na eventualidade de ser realizado em grupo, cada elemento do grupo deverá controlar a totalidade do projeto;
- O trabalho deverá ser entregue em duas fases. Cada fase do trabalho realizado deverá ser submetido num ficheiro compactado, na plataforma a indicar, cujo nome deverá conter o número do aluno e número da fase. No caso de o trabalho ser feito em grupo, devem constar os dois números dos alunos desse grupo.

Exemplo para a entrega da Fase 1 do aluno 1234: *trabalhoPOO_1234_Fase1.zip*

As datas de cada fase e as metas a atingir em cada uma, são:

- **Fase 1 (15-11-2023)**
 - Diagrama de classes
 - Implementação essencial das classes
 - Definição das estruturas de dados a utilizar
 - Estruturação do Projeto em camadas
 - Relatório do trabalho desenvolvido até à data
- **Fase 2 (19-12-2023)**
 - Implementação final das classes e serviços
 - Aplicação demonstradora dos serviços implementados
 - Relatório final do trabalho realizado

6 Problemas a explorar

O problema a explorar é do critério do aluno. Mostram-se contudo alguns exemplos de problemas que podem ser explorados:

- Sistema que permita a gestão de ocorrências de incêndios florestais e outras catástrofes naturais;

- Sistema que permita gerir uma urgência de um hospital: listas de espera, prioridades, etc...;
- Sistema que permita gerir pessoas infetadas numa situação de crise de saúde pública: registar novos casos, contabilizar casos, consultar casos por região, sexo, idades, outros. Etc.
- Sistema capaz de gerir um parque de estacionamento de um Hóspital, com múltiplos setores de estacionamento.

7 Critérios de Avaliação

A avaliação neste trabalho prático contempla as seguintes partes:

| Critérios de Avaliação | |
|-----------------------------------|-----|
| Qualidade do Relatório produzido | 20% |
| Qualidade da solução desenvolvida | 50% |
| Defesa individual final | 30% |

Considerando que:

7.1 Qualidade do Relatório

- Estrutura, Clareza e Expressividade
- Capacidade de síntese e correção da escrita

7.2 Qualidade da solução desenvolvida

- Qualidade do código produzido: estrutura da solução, nome de ficheiros, uso de bibliotecas, Norma CLS;
- Organização e Implementação das Classes;
- Qualidade dos algoritmos aplicados;
- Estruturas de dados exploradas;
- Tratamento adequado de exceções;
- Persistência de dados com recurso a ficheiros ou outros;
- Exploração de outras valências: Lambda Functions; LINQ; MVC; WPF;
- Qualidade da API produzida;
- Cumprimento dos prazos e fases estipuladas.

Bom trabalho.

lufer