

# INTRODUÇÃO À LINGUAGEM ORIENTADA A OBJETOS

## PROGRAMAÇÃO E SISTEMAS DE INFORMAÇÃO – 2º ANO CURSO PROFISSIONAL TÉCNICO DE GPSI



*“Determinação, coragem e autoconfiança são fatores decisivos para o sucesso. Se estamos possuídos por uma inabalável determinação, conseguiremos superá-los. Independentemente das circunstâncias, devemos ser sempre humildes, recatados e despidos de orgulho.”*

(Dalai Lama)



### Módulo 09

#### Introdução à Linguagem Orientada a Objetos



### Introdução

A Programação Orientada por Objetos (POO) caracteriza-se, fundamentalmente, pela criação de classes que, depois, são usadas nos programas sob a forma de objetos.

As principais características da POO são o encapsulamento, a herança e o polimorfismo. No encapsulamento, as classes e os objetos são unidades de código que encobrem e protegem os seus membros em relação a acessos externos. Na herança, as classes podem ser criadas com base em outras classes, herdando os seus membros. No polimorfismo, um determinado método de uma classe ou conjunto de classes pode ter várias (poli) versões ou formas (morfismo) para funcionar em diferentes contextos.



### Recursos

- Documentação C#  
<https://learn.microsoft.com/pt-pt/dotnet/csharp/>

## Ferramentas

- Flowchart Maker & Online Diagram Software  
<https://app.diagrams.net/>
- Suporte para diagrams.net  
<https://github.com/jgraph/drawio/wiki/Getting-Support>



## Desafios

O conceito de **portefólio** está subjacente à ideia de compilação e exposição de todo o trabalho desenvolvido numa determinada área. O portefólio digital pode ser uma ferramenta de trabalho bastante prática que permite organizar todo o material que é utilizado e desenvolvido ao longo do módulo da disciplina de PSI.

**O teu portefólio servirá como um dos instrumentos de avaliação, no domínio “COMUNICAR E COLABORAR”.** No final do módulo, terás a oportunidade de apresentar e defender o portefólio das tuas práticas, como programador, com o objetivo de tornar evidente a qualidade do teu trabalho. Como a avaliação é um processo dinâmico e necessita de reflexão, o teu portefólio deve armazenar todos os programas desenvolvidos e toda a informação pertinente que reflita o teu empenho e o teu trabalho autónomo, facilitando assim, a tua reflexão sobre todo este processo.

Os problemas apresentados estão **organizados e agrupados por níveis de esforço e de complexidade** na sua resolução.

Desenvolve as tão faladas competências específicas para o século XXI e abraça este desafio como uma oportunidade de aprendizagem. Colabora na plataforma Teams da disciplina para partilhares o teu conhecimento com os colegas de equipa. Aplica o que aprendestes para resolver novos problemas e transfere o que sabes para os teus colegas na plataforma Teams. Desenvolve programas e soluciona o maior número de problemas para enriqueceres o teu portfólio e, assim, estares preparado para seres um programador atuante num mundo em transformação.

Dos casos práticos que são apresentados, pretende-se encontrar uma solução computacional que devolva ao utilizador o resultado esperado. Na perspetiva da programação orientada por objetos, especifique as classes que podem definir a interface e a implementação dos objetos dos problemas descritos. Para a(s) classe(s) identificada(s), deve apresentar os estados (propriedades) e os comportamentos (métodos) mais adequados.

Antes de escrever o código é importante analisar e desenhar o Diagrama de Classe que define a interface do problema. O diagrama de classes pode ser desenhado numa aplicação web e o código deve ser desenvolvido em C#, num IDE à tua escolha.

Mas prepara-te, porque vamos também desafiar-te a resolver problemas com um maior nível de complexidade, para elevares e construíres o teu próprio conhecimento.



1. Desenvolva uma classe chamada **Quadrado** para representar um quadrado, ou seja, um quadrilátero regular com quatro lados com o mesmo comprimento. A classe possui um único atributo denominado **lado**, do tipo **double**, que representa o lado do quadrado e cujo valor deve ser maior ou igual a zero e menor ou igual a vinte. A classe possui dois construtores: o primeiro configura o lado do quadrado com o valor padrão 1.0, e o segundo recebe como parâmetro o lado do quadrado. O lado do quadrado pode ser obtido e alterado pelo utilizador por meio de métodos. A classe também apresenta os métodos **area()**, **diagonal()** e **perimetro()**, que devolvem a área, a diagonal e o perímetro do quadrado, respetivamente. A área de um quadrado de lado  $l$  é obtida pela fórmula  $2 * l$ . A diagonal de um quadrado de lado  $l$  é obtida pela fórmula  $\sqrt{2} * l$ . O perímetro de um quadrado de lado  $l$  é obtido pela fórmula  $4 * l$ .
2. Crie uma classe para representar um jogador de futebol, com os atributos nome, posição, data de nascimento, nacionalidade, altura e peso. Crie os métodos públicos necessários para leituras e escritas e também um método para imprimir todos os dados do jogador. Crie um método para calcular a idade do jogador e outro método para mostrar quanto tempo falta para o jogador se aposentar. Para isso, considere que os jogadores da posição de defesa se aposentam em média aos 40 anos, os jogadores de meio-campo aos 38 e os atacantes aos 35.
3. Crie uma classe chamada **Bilhete**, que possui um valor em euros e um método **imprimirValor()**. Crie uma classe **BilheteVIP**, que herda da classe **Bilhete** e possui um valor adicional. Crie um método que retorne o valor do bilhete VIP (com o valor adicional incluído). Crie um programa para criar as instâncias de **Bilhete** e **BilheteVIP**, mostrando a diferença de preços.
4. Crie uma classe **Elevador** para armazenar as informações de um elevador de um prédio. A classe deve armazenar o andar atual (pisos = 0), total de andares no prédio (desconsiderando o piso 0), capacidade do elevador e quantas pessoas estão dentro dele. A classe deve também disponibilizar os seguintes métodos:
  - **Iniciar()**: que deve receber como parâmetros a capacidade do elevador e o total de andares no prédio (os elevadores começam no piso e vazio);
  - **Entrar()**: para acrescentar uma pessoa no elevador (só deve acrescentar se ainda houver espaço);
  - **Sair()**: para remover uma pessoa do elevador (só deve remover se houver alguém dentro dele);
  - **Subir()**: para subir um andar (não deve subir se já estiver no último andar);
  - **Descer()**: para descer um andar (não deve descer se já estiver no térreo);

Obs.: Encapsular todos os atributos da classe.



5. Crie uma classe **Produto** para representar um produto do mundo real. A sua classe deverá conter os seguintes atributos e métodos:
- Um campo de dados privado do tipo **String** chamado **nome**, que representará o nome do produto.
  - Um campo de dados privado do tipo **double** chamado **precoCusto**, que guardará o preço de custo do produto.
  - Um campo de dados privado do tipo **double** chamado **precoVenda**, que guardará o preço de venda do produto.
  - Um campo de dados privado do tipo **double** chamado **margemLucro**, que guardará a margem de lucro do produto.
  - Crie os métodos necessários para aceder e modificar os atributos da classe **Produto**, tendo em atenção que o preço de venda não pode ser inferior ao preço de compra. Caso isso aconteça, exiba uma mensagem alertando o utilizador.
  - Crie um método chamado **calcularMargemLucro()** que calculará a margem de lucro do produto.
  - Crie um método chamado **getMargemLucroPorcentagem()** que retornará a margem de lucro como percentagem.
  - Para finalizar, no método **main()** da classe **Principal**, crie um novo objeto da classe **Produto**, peça para o utilizador informar os preços de custo e de venda e exiba a margem de lucro em moeda e em percentagem. A sua saída deverá ser algo parecido com o mostrado abaixo:

```
Digite o preço de custo: 120
Digite o preço de venda: 195
Preço de custo: 120.0
Preço de Venda: 195.0
Margem de Lucro: 75.0
Margem de Lucro em Percentagem (%): 62.5
```

6. Um banco possui vários tipos de contas bancárias: (1) conta-corrente simples; (2) conta-corrente especial; (3) conta poupança.

Todas as contas possuem um número, um titular e um saldo. Para cada uma delas, entretanto, existem peculiaridades.

Conta corrente simples: tem direito a um cartão de débito (guardar o número) e um talão de cheques (guardar o número do primeiro e do último cheque do talão), mas não tem direito a limite e nem cheque especial.

Conta corrente especial: tem tudo o que a conta simples possui, mais o limite de crédito concedido pelo banco e taxa de juros cobrada pelo uso do limite.

Conta poupança: tem tudo o que a conta simples possui, mais uma data de aniversário (o dia do mês em que os rendimentos são creditados).

Defina as classes anteriores, utilizando herança onde for apropriado.

7. Crie uma aplicação que gere uma conta de cada tipo e registe (nos métodos construtores) as informações pertinentes para cada uma das contas.

Depois que as contas estiverem registadas, o seu programa deverá mostrar um menu com opções para visualização das contas e para atualização do saldo de cada conta, lembrando que: para as contas simples, o saldo está sempre atualizado; para as contas especiais, o valor do limite utilizado deverá ser subtraído do saldo, e, para as contas poupanças, o saldo deverá ser acrescido do rendimento (vamos supor 2% ao mês).

8. Um promotor deseja controlar todos os eventos dos quais participa e você foi contratado para resolver esse problema.

Assim que começou o trabalho, você ficou a saber que o registo de um evento necessita de: um identificador único (um código), descrição, local e data de realização, quantidade de convites colocados à venda, custos da organização e valor da entrada.

Foi-lhe informado, também, que existem sempre três valores de entrada: o 1º valor é para as entradas do tipo popular, o 2º valor é para as entradas do tipo normal e o 3º valor é para as entradas do tipo VIP.

Alguns destes eventos são festas bar aberto, possuindo, assim, além de todas as informações anteriores, uma relação das quatro bebidas que serão servidas. As informações da bebida, por sua vez, são o nome, teor alcoólico e valor unitário.

O valor das entradas populares pode ser informado pelo utilizador ou pode ser calculado automaticamente, da seguinte forma:

- a) eventos bar aberto: 20 € mais 50% do valor unitário de cada bebida que será servida;
- b) demais eventos: custo de organização dividido pela quantidade de convites colocados a venda.

O valor da entrada normal é o valor da popular + 10% desse mesmo valor. O valor da entrada VIP é o valor da normal + 15% desse mesmo valor.

Implemente as classes que representem este contexto. Utilize, onde apropriado, todos os conceitos de orientação a objetos já estudados.

Crie, também, uma classe aplicação que permita o registo de cinco eventos (que podem ser Bar aberto ou não, dependendo do utilizador).

Durante o registo, deverão ser calculados, nas classes apropriadas, os valores das entradas, de acordo com o desejo do utilizador e seguindo as regras de negócio, descritas anteriormente.

Mostre, quando o utilizador desejar, todas as informações de todos os eventos registados.



### 9. Simulação Balão de Ar

Um pequeno programa simula o “voo” de um balão. Cada balão tem uma cor, a sua direção atual e a altura a que se encontra do solo. Durante o seu trajeto é preciso saber o estado do balão a cada momento, mudar a sua cor, mudar a sua direção e controlar a sua altura. Note-se que o balão pode subir até ao infinito, mas descer só pode ser até à altura 0.

### 10. Clube de Vídeo

Um clube de vídeo aluga DVD's aos seus clientes. Para cada DVD é necessário saber o código do filme, nome, duração, realizador, origem e tipo de filme. Os clientes são caracterizados pelo código, nome, morada, data de nascimento, data de adesão. Os clientes podem ser privados ou empresas.

### 11. Alunos de Licenciatura da Universidade

Os alunos de licenciatura da universidade, sejam diurnos ou noturnos, são identificados por meio de um número mecanográfico, o qual é atribuído de forma crescente. Para o caso dos alunos noturnos interessa saber a empresa onde eventualmente trabalham, a posição que ocupam na hierarquia da empresa e a experiência profissional que possuem.

Existem também alunos de mestrado, sendo estes igualmente identificados por um número mecanográfico, atribuído de forma crescente e de modo independente dos de licenciatura.

Tanto os alunos de mestrado como os de licenciatura possuem nome, morada e telefone.

### 12. Simulação Autocarro/Passageiros

Um pequeno programa simula o fluxo de passageiros durante uma viagem de autocarro de determinada empresa. A viagem de autocarro tem uma rota predefinida com paragem em certas cidades. As paragens possíveis serão necessariamente um subconjunto das cidades onde a empresa opera. Neste caso, suponhamos que essas cidades são Lisboa, Coimbra, Aveiro, Porto e Braga.

Nesta simulação os passageiros são entidades anónimas. Apenas é necessário saber a sua paragem de origem e a sua paragem de destino. Quanto aos autocarros, estes têm um determinado número de lugares e uma rota definida por uma lista ordenada de paragens, sendo a primeira o ponto de partida e a última o destino final. Deverá ser possível saber quantos passageiros estão dentro do autocarro, qual a última paragem efetuada, e quantos passageiros vão sair em determinada paragem.

A simulação deverá poder ser executada com base num conjunto de passageiros (com origem/destino) e um autocarro (com a rota definida). O autocarro segue de paragem em paragem, sendo que em cada paragem saem e entram passageiros de acordo com a sua origem/destino. A simulação deverá emitir mensagens sobre cada paragem, nomeadamente, quantos passageiros saíram, quantos entraram, e quantos não tiveram lugar para viajar.

### 13. Simulação Cartão de Cliente

Um cartão de cliente é um cartão que acumula pontos de bonificação à medida que são registadas compras e que possui o valor total em dinheiro das compras realizadas, um código alfanumérico e um nome de titular. O titular do cartão tem nome, telefone, email e cartão de cidadão.

Num dado estabelecimento, as regras são as seguintes: por cada compra efetuada em euros, o cliente recebe de bónus um número de pontos que é o arredondamento para baixo de 10% do valor da compra. Sempre que é atingido um valor múltiplo de 50 pontos, o cliente acumula mais 5 pontos por cada 50, que são acrescentados aos que já tinha no cartão.

O cartão permite ainda descontar pontos ao cartão devido a levantamento de um prémio, modificar o titular do cartão, modificar a taxa de prémio, passando por exemplo de X% para Y% e inserir no cartão a informação de uma nova compra de certo valor e atualizar dados.

### 14. Simulação Aeródromo do Caramulo

O aeródromo do Caramulo, precisa de uma aplicação para registar os aviões estacionados, os seus donos e os veículos de apoio. Foi feito o levantamento dos requisitos necessários à aplicação tendo-se obtido as seguintes informações:

- Os aviões são agrupados por categorias (tipos de avião) por razões relativas aos regulamentos da aviação civil. Utilizam o aeródromo aviões de combate a incêndio (água ou espuma) e de transporte de passageiros (nº de lugares).
- Cada avião tem um número de registo (atribuído uma única vez e que não pode ser alterado), número de horas de voo, pertence a um determinado tipo de aviões e tem um lugar fixo num determinado hangar.
- Cada hangar existente no aeródromo é designado por um número, tem uma dada capacidade e uma localização.
- Um avião pode pertencer a uma pessoa singular ou a uma empresa.
- Possui também veículos ligeiros, bombeiros e autocarros.

**BOM TRABALHO! TU ÉS CAPAZ! CONSTRÓI O TEU CONHECIMENTO...**

Os facilitadores da aprendizagem,

*Andreia Quintal | Carlos Almeida*