

Nome:

Nº Mec.

---

**Notas importantes:**

1. Crie um projeto no seu computador com o seu número mecanográfico.
2. Descarregue os ficheiros de base para esta prova a partir do link de submissão existente no elearning “Testes (ATP1, ATP2, AP) > Teste Prático 2025 - !! Só Enunciado !!”
3. No final do teste, aceda de novo ao *elearning* e, use o link para submissão para enviar o código do projeto compactando-o num único ficheiro com nome igual ao seu número mecanográfico.

## Gestão de Concertos de uma Banda

Escreva um programa que modele um sistema de gestão de concertos de uma banda. Cada concerto é representado por um objeto Concert. O programa deve usar Java Collections para gerir a faturação desses concertos.

Cada Concert deve conter os seguintes atributos:

- Identificador único (int) – deve ser automático, incremental a começar em um (i.e., 1)
- Duração (double) – unidade é minuto
- Local do concerto (String) – formato “Cidade, País”
- Data e hora do início (LocalDateTime)

**Informação adicional:**

```
DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm");  
LocalDateTime date = LocalDateTime.parse("2025-05-29 16:00", formatter);  
System.out.println(date.format(formatter));
```

O programa deve incluir as seguintes classes e interfaces: **[modelação 15%]**

- **Concert**: é a classe principal que representa um concerto. Esta classe deve ter os atributos mencionados acima, um construtor, função `toString()` e os `setters` e `getters` apropriados.
- **IConcertProfitCalculator**: Uma interface que define o comportamento dos objetos (i.e., calculadoras) que permitem projetar o lucro de cada concerto. Deve conter o seguinte método:
  - `calculateConcertProfit(Concert c)`: Retorna o lucro estimado para o concerto “c”.
  - Pode usar o ficheiro `IConcertProfitCalculator.java` e fazer as alterações que achar necessárias.
- **StandardConcertProfitCalculator**: **[10%]** Uma classe que implementa a interface **IConcertProfitCalculator**. O custo dessa transação é calculado da seguinte forma:
  - Lucro base de **€1500/h** para todos os concertos.

- Duplica o lucro sempre que o concerto seja fora de Portugal.
- Aumenta o custo em 800€ (fixo) sempre que o concerto for fora da península ibérica.
- **ConcertManager:** Uma classe que gere a faturação dos concertos. A classe deve usar Java Collections para guardar e gerir as transações. A classe deve oferecer os seguintes métodos:
  - **addConcert**(Concert c): Adiciona um novo concerto ao sistema. **[5%]**
  - **removeConcert**(int id): Remove um concerto usando o identificador único. **[5%]**
  - **getConcert**(int id): Obtém o concerto no sistema com base no identificador. Retorna o concerto ou null se o concerto não existir. **[5%]**
  - **calculateConcertProfit**(int id): Calcula o lucro de fazer esse concerto usando a StandardConcertProfitCalculator. No caso desse concerto não existir, retorna -1. **[10%]**
  - **printAllConcerts**(): Imprime a informação de todas as transações no sistema. **[10%]**
  - **sortConcertsByProfit**(): imprime no ecrã todos os concertos **separados por mês do ano**. Dentro de cada mês, os concertos devem ser impressos por **ordem decrescente** do seu lucro. **[15%]**
  - **readFile**(String fich): Importa a informação dos concertos a partir de um ficheiro. **[10%]**
    - Nota #1: se preferir pode substituir este método por um construtor.
    - Nota #2: se o ficheiro contiver um concerto com um identificador já existente, esse concerto é substituído pelo que está no ficheiro.
    - Nota #3: assegure-se que o identificador único automático e incremental se mantém consistente com o identificador importado com valor mais elevado.
  - **writeFile**(String fich): Escreve a lista de concertos existentes no ConcertManager para um ficheiro. O ficheiro deve ter os dados separados por “;” e incluir em cada linha a seguinte informação para cada concerto: **[15%]**
    - Identificador único
    - Duração
    - Local do concerto
    - Data e hora do início
    - Lucro do concerto

**Utilize o ficheiro ConcertTester.java para testar e demonstrar o uso da classe ConcertManager (não precisa de menu).** Pode editar este ficheiro livremente, mas deverá assegurar-se que este continua a testar as seguintes funcionalidades:

- Cria uma instância de ConcertManager.
- Adiciona concertos ao sistema.
- Remove concertos do sistema.
- Lê a partir de um ficheiro os dados referentes a vários concertos.
  - i.e, o programa deve conseguir ler o ficheiro classicpimba.txt
- Obter um concerto específico através do seu identificador.
- Imprime a informação de todos os concertos registados no sistema.
- Calcula o lucro de um concerto.
- Imprime no ecrã a lista de concertos separados por mês, ordenados por lucro (decrescente).
- Escreve para um ficheiro a lista de concertos existentes no ConcertManager.