Exercício 12.3 – Gestão de Tarefas (revisões)

Escreva um programa que permita gerir uma lista de tarefas pendentes. A classe Task permite representar cada tarefa que irá fazer parte da lista. O programa deve usar **Java Collections** para gerir as tarefas.

Cada tarefa deve conter os seguintes atributos:

- Identificador único (int) – deve ser automático e incremental
- Título da Tarefa (String)
- Descrição (String)
- Data Limite (LocalDate)
- Prioridade (int) – valores de 1 (mais baixa) a 5 (mais alta)
- Estado (String) – valores possíveis: "Pendente", "Em Progresso", "Concluída"

O programa deve conter, pelo menos, as seguintes classes:

1. Task

Classe que representa uma tarefa.

Atributos:

- id (int)
- title (String)
- description (String)
- dueDate (LocalDate)
- priority (int)
- status (String)

Métodos:

- Construtor com todos os campos exceto o id (gerado automaticamente)
- Métodos getters e setters
- Método toString()

---

2. TaskManager

Classe responsável por gerir a lista de tarefas. Deve implementar os seguintes métodos:

- addTask(Task t): Adiciona uma nova tarefa ao sistema. Não permita adicionar uma tarefa duplicada (título + data + prioridade).
- removeTask(int id): Remove uma tarefa com base no seu identificador.
- getTask(int id): Retorna uma tarefa com o id indicado, ou null se não existir.
- updateStatus(int id, String newStatus): Atualiza o estado da tarefa.
- getTasksByPriority(int priority): Retorna a lista de tarefas com a prioridade especificada.
- getTasksDueAfter(LocalDate date): Retorna as tarefas com data posterior à indicada, utilizando **Streams**.
- printAllTasks(): Imprime todas as tarefas.
- readFile(String filename): Lê tarefas de um ficheiro de texto (ex.: tarefas.txt) e adiciona ao sistema.
- writeFile(String filename): Escreve todas as tarefas atuais para o ficheiro, separadas por ;.

---

3. TaskCostCalculator

Classe responsável por calcular o "custo" de uma tarefa com base na sua prioridade:

- Prioridade 1 ou 2 → 1.00€
- Prioridade 3 ou 4 → 2.50€
- Prioridade 5 → 5.00€

Esta classe deve implementar uma interface ITaskCostCalculator com o método:

- double calculate(Task t)

---

4. TaskTester.java

Classe que testa as funcionalidades principais. Deve:

- Criar uma instância de TaskManager
- Adicionar tarefas

- Editar tarefas
- Remover tarefas
- Carregar tarefas de um ficheiro tarefas.txt ao iniciar
- Exportar para tarefas.txt ao terminar
- Testar cálculo de custo de uma tarefa
- Listar todas as tarefas
- Filtrar tarefas com dueDate > 01/01/2025

Extras obrigatórios:

- Valide a data de vencimento (não pode ser no passado).
- Valide o estado da tarefa (apenas valores permitidos).
- Valide que a prioridade esteja entre 1 e 5.
- Implemente a igualdade de tarefas baseada no título, data e prioridade.