ULHT
Linguagens de Programação II
LIG / LEI / LEIRT
2020/2021
Bruno Cipriano, Rodrigo Correia, Lúcio Studer, Pedro Alves
v1.0.0

Segundo exercício de modelação, implementação e testes automáticos

Introdução

Com este exercício pretende-se que os alunos pratiquem a modelação e implementação de sistemas usando Programação Orientada por Objectos e a linguagem Java.

Os alunos devem analisar a descrição da realidade que se pretende modelar, de forma a:

- Definir o modelo de classes respectivo, usando UML;
- Implementar o seu modelo, usando Java;
- Criar alguns testes automáticos, usando JUnit.

Nota: este exercício dá continuidade ao exercício anterior (intitulado “Exercício de Modelação, Implementação e Testes Automáticos”).

Domínio

A Empresa **XPTO, S.A.**, tendo ficado satisfeita com os trabalhos desenvolvidos pelos alunos da ULHT no seu pedido anterior, decidiu voltar a recrutar os serviços dos mesmos para o desenvolvimento de novas funcionalidades para o seu sistema.

O objectivo é estender o sistema existente de forma a dar resposta aos seguintes novos requisitos:

- A empresa passou a ter vários tipos de funcionários (que se descrevem mais abaixo), sendo que cada tipo de funcionário tem uma forma diferente de cálculo de salário.

- A empresa passou a querer ter relatórios sobre os clientes que encomendam cada tarefa e sobre o preço que os clientes pagam pelos serviços realizados pela empresa.

Seguem-se informações mais pormenorizadas sobre os novos conceitos e funcionalidades que se pretendem implementar

Clientes

Um cliente pode ser uma empresa (cliente empresarial) ou uma pessoa (cliente individual).

Um cliente é caracterizado por:

- Nome
- Contacto telefónico
- NIF

Os **clientes empresariais** têm informação extra:

- Nome pessoa responsável
- Telefone pessoa responsável

Tipos de funcionários

Com o crescimento da empresa, a mesma sentiu a necessidade de rever a sua estrutura organizacional.

Desta forma, passaram a existir três tipos de funcionários:

- Gestor
- Tarefeiro
- Técnico de suporte informático

Tipo	Descrição
Gestor	O Gestor é dono de um conjunto de tarefas, sendo responsável por interagir com o cliente. O Gestor não executa as tarefas pelas quais é responsável.
Tarefeiro	O Tarefeiro executa tarefas. (Cada tarefa continua a ser executada apenas por um Tarefeiro.)

Tipo	Descrição
Técnico de suporte informático	Realiza trabalhos internos da própria empresa.

Salário do Gestor

Os gestores têm um salário base, cujo valor (em unidades monetárias) varia de gestor para gestor.

Para além disso, recebem um bónus de 0.5 unidades monetárias por cada tarefa cuja gestão seja feita por sí.

A fórmula usada é então a seguinte:

```
Salário = Salário Base + 0.5 * Nr. De Tarefas Geridas
```

(Dica: é necessário que o modelo preveja uma forma de alimentar o salário base de cada gestor criado no sistema.)

Salário do Tarefeiro

Os tarefeiros, para além de receberem um salário base (de 505 unidades monetárias), recebem também um valor por cada hora de trabalho. Podem ainda receber um bónus.

Para compensar os tarefeiros que tenham feito **muitas tarefas (mais do que 100) e poucas horas** (menos do que **75**), a empresa paga-lhes um bónus de 0.25 por cada tarefa acima das 50. Os tarefeiros que não se enquadrem na situação descrita recebem **zero** de bónus.

A fórmula usada para calcular o salário mensal de cada tarefeiro é a seguinte:

```
Salário = 505 + 2 * NúmeroDeHorasTrabalhadas + Bónus
```

Técnico de suporte informático

Os técnicos de suporte têm um salário fixo que pode variar de técnico para técnico.

(Dica: é necessário que o modelo preveja uma forma de alimentar o salário base de cada técnico criado no sistema.)

Custo do serviço

Os clientes pagam, por cada tarefa encomendada, o seguinte preço:

```
Preço = 500 + 10 * Nr. Horas Tarefa
```

Funcionalidades

Para além das funcionalidades (relatórios) implementadas originalmente, a empresa pretende que o sistema lhe permita obter os seguintes relatórios:

- Relatório com a lista de clientes
    - Deve incluir os dados completos do cliente

- Relatório com todas as tarefas encomendadas por cada cliente
    - Para cada cliente, deve aparecer a descrição das tarefas que encomendou.

- Relatório com todas as tarefas encomendadas por cada cliente num determinado mês de calendário (p.e. Quais as tarefas encomendadas pelo cliente ZYX no mês de Maio?).
    - Deve incluir o preço de cada tarefa e o preço total a pagar pelo cliente.

(Por efeitos de simplificação, pode assumir que as tarefas nunca se estendem para além de um mês de calendário.)

(Os conteúdos dos relatórios devem ser apresentados no ecrã.)

Exercícios

1) Defina um modelo de classes que permita responder aos requisitos acima mencionados e represente esse modelo usando UML;

2) Implemente no seu projecto as classes Java e os métodos necessários para obter os relatórios indicados;

3) Identifique três casos de teste para o cálculo de salários **de cada tipo** de funcionários.

Apresente esses casos em forma de tabela. A sua tabela deve indicar: estado inicial, a acção realizada e o resultado e/ou estado final esperados.

(Dica: Tenha em conta os casos limite.)

3.1) Justifique as escolhas que fez para definir cada um dos casos de teste.

3.2) Implemente, usando JUnit, os casos de teste definidos nos passos anteriores.