
PROGRAMAÇÃO ORIENTADA AOS OBJECTOS

(2º ANO – LEI + LCC)

TRABALHO PRÁTICO OBRIGATÓRIO - 6/6/2010

PARTE II: Entrega Electrónica 6 de Junho de 2010

GESTÃO DE SEGUROS DE UMA EMPRESA SEGURADORA

Definidas que foram na parte I as classes de seguros e suas características principais, cada empresa seguradora possui o registo dos seus clientes, o registo dos seguros que pode vender e o registo dos seguros já vendidos.

Sobre cada cliente pretende-se ter a seguinte informação: código, nome, morada, BI, NIF e informação sobre o conjunto de seguros de que é titular.

Sobre cada seguro efectivamente vendido a um cliente, pretendemos, para além da informação indicada na parte I, ter também os seguintes dados: data de início (todos são anuais), pago ou em atraso e forma de pagamento (cf. \$\$, transferência bancária, etc).

Sobre toda esta estrutura de informação pretende-se agora implementar as seguintes operações:

- Determinar o número actual de clientes que são titulares de seguros;
- Determinar se a seguradora, em comparação com os números do último ano, está a ganhar ou perder clientes;
- Determinar a colecção de todos os tipos de seguro oferecidos;
- Para um qualquer seguro determinar o seu prémio actual (valor a pagar nesse momento);
- Determinar os códigos de todos os seguros com pagamento em atraso;
- Criar um novo seguro de um tipo existente para um dado cliente que não tenha seguros em atraso;
- Remover cliente (apenas se não for titular de nenhum seguro);
- Extinguir um dado seguro de um dado cliente (usando os respectivos códigos);
- Determinar o titular do seguro mais caro;
- Procurar e devolver a ficha completa do cliente X (como String);
- Colecção dos códigos de todos os clientes titulares de um dado tipo de seguro;
- A seguradora está cotada em Bolsa: implemente para a seguradora a interface Cotavel definida como: `double getValorBolsa(); void setValorBolsa(double); void setValInicio(double)`

Todos os objectos da aplicação devem poder ser guardados em ficheiro sob a forma de ObjectStreams. O estado global do sistema deve poder ser escrito e lido de ObjectStreams.

Nota: Quando tiver que apresentar presencialmente o projecto, garanta que a aplicação arranca desde o início com um estado representativo, ou seja, um número razoável de clientes, seguros, etc. No momento de apresentação do projecto não são permitidas criação de instâncias de qualquer das classes do projecto, por exemplo, em BlueJ no momento. Tudo já deve estar construído e disponível para teste dos métodos.

Serão testados todos os métodos acima apresentados. Tenha em atenção, sempre que possível, que listas de códigos, de nomes, etc., devem surgir ordenadas.

Prof. F. Mário Martins