UNIVERSIDADE ABERTA
www.uab.pt

Ministério da Educação e Ciência

e-Fólio

U.C. 21077

Linguagens de Programação e-Fólio B – Linguagem Prolog

-- INSTRUÇÕES --

1) O e-fólio tem uma cotação de 4 valores.

2) Qualquer tentativa de plágio resultará numa nota final de zero valores.

3) Este e-fólio deve ser resolvido usando a *linguagem Prolog*.

4) Deve ser submetido um ficheiro comprimido (ZIP ou RAR) com o nome e número de estudante contendo:

a) Código do programa;

b) Ficheiro readme.txt com a informação necessária para compilar e executar o programa;

c) Relatório até 4 páginas descrevendo a solução apresentada e os testes efetuados.

5) Serão considerados os seguintes critérios de avaliação:

a) 40% C1 –Resolução do Problema

b) 40% C2 –Qualidade do Código

c) 20% C3 –Documentação (Relatório + Readme)

E-fólio B

Como recém-contratado no setor de TI de um Banco, foi-lhe designada a tarefa da criação de um sistema de gestão bancária, para uso interno dos gestores de conta. O seu Diretor de TI insiste que se deva usar a linguagem *Prolog* no back-end pois considera que a mesma é mais veloz no processamento de informação, e instruiu que o front-end do sistema será em modo consola feito em *Java*. Após reunir com alguns futuros utilizadores e com os arquitetos de sistemas, os seguintes requisitos foram levantados:

Atualmente, os Gestores de conta usam tabelas para controlo de contas, da seguinte forma:

Informações de cliente:

Número de Cliente	Nome	Agência	Cidade	Data de Abertura
123	Alice	NYC-123	New York	01-01-2020
456	Bob	CHI-456	Chicago	02-02-2020
789	Charlie	LA-789	Los Angeles	03-03-2020
752	John	CHI-456	Chicago	03-03-2020

Balanço total da conta: (Representa o valor total do balanço corrente mais o valor disponível de crédito)

Número de Cliente	Balanço Total (Currente+Crédito)
123	2500
456	500
789	50
752	5000

Balanço de crédito:

Número de Cliente	Balanço Crédito
123	0
456	5000
789	2000

Movimentos:

Número de Cliente	Valor(+-)	Data do Movimento
123	-100	01-01-2020
456	200	02-02-2020
789	-10	03-03-2020
789	-20	04-03-2020

Para sua facilidade o sistema apenas considera números inteiros nos seus valores.

1. Os arquitetos de sistema, propuseram que você mantenha esta estrutura, a quando na criação dos seus factos, pois o utilizador já está habituado a ela. Crie os factos necessários para o seu programa.

2. Obtenha todos os seus clientes.
    a. Crie um predicado que obtém todos os dados dos seus clientes em Prolog.
    b. Crie uma classe Cliente com os atributos necessários de um cliente Número de Cliente, Nome, Agência, Cidade e Data de Abertura.

c. Crie Uma Classe SistemaBancario que armazena uma lista, vetor ou matriz do tipo da classe Cliente, para salvar os seus clientes, o construtor da classe deve obter do Prolog a lista de clientes e salvar cada um na lista, vetor ou matriz e ordenar a lista por ordem crescente de Número de cliente.
d. Crie um método na classe SistemaBancario para imprimir a lista de clientes com os dados de cada cliente.
e. Crie um método na classe SistemaBancario que apresenta um menu de opções onde a primeira opção é mostrar a lista de clientes e a última é sair do programa.

3. Obtenha os clientes com base na cidade.
   a. Crie um predicado em Prolog que obtenha o número de cliente e o nome de cliente com base numa cidade.
   b. Crie um método na classe SistemaBancario que obtenha do Prolog a lista de clientes para uma determinada cidade e imprima essa lista com o número de cliente e nome.
   c. Adicione ao menu previamente criado a opção de ver clientes de determinada cidade.

4. Verificar clientes elegíveis a crédito. Um cliente é elegível a crédito se o seu saldo no balanço total for superior a 100 unidades monetárias e o seu crédito for 0 ou inexistente.
   a. Crie um predicado no prolog que obtenha todos os clientes elegíveis a crédito, deverá retornar todos os dados das informações de cliente, Número de Cliente, Nome, Agência, Cidade e Data de Abertura.
   b. Crie um método na classe SistemaBancario que obtenha do Prolog todos os clientes elegíveis a crédito e imprima todos as suas informações de cliente.
   c. Adicione ao menu previamente criado a opção de ver clientes elegíveis a crédito.

5. Aceder a operações de cliente.
   a. Crie um método na Classe Cliente que apresente um menu de cliente, inicialmente com apenas a opção voltar ao menu anterior.
   b. Crie um método na classe SistemaBancario onde seja possível selecionar um cliente da sua lista, vetor ou matriz de clientes, com base num determinado número de cliente, e chamar o método de menu desse cliente (Classe Cliente).
   c. Adicione ao menu previamente criado (da Classe SistemaBancario), o método criado anteriormente, ou seja, a opção de selecionar um determinado cliente pelo seu número e ver o seu menu de cliente.

6. Verificar o saldo real do cliente. O saldo Real do cliente é dado pela diferença do balanço total e o balanço de crédito.
   a. Crie um predicado em Prolog que obtém o saldo real de um determinado cliente.
   b. Crie um método na classe Cliente que recorra ao Prolog para obter o saldo real do cliente e o imprima.
   c. Adicione ao menu de cliente previamente criado a opção para ver saldo de cliente.

7. Verificar o Balanço de crédito do cliente.
   a. Crie um predicado em Prolog que obtém o balanço de crédito de um determinado cliente.
   b. Crie um método na classe Cliente que recorra ao Prolog para obter o balanço de crédito do cliente e o imprima.
   c. Adicione ao menu de cliente previamente criado a opção para ver o balanço de crédito.

8. Ver movimentos de cliente.
   a. Crie um predicado em Prolog que obtenha os movimentos de determinado cliente deve retornar os movimentos com o valor e a data.
   b. Crie um método na classe Cliente que recorra ao Prolog para obter os movimentos e os imprima com o seu valor e data.

c. Adicione ao menu de cliente previamente criado a opção de ver os movimentos do cliente.

9. Fazer depósitos e levantamentos. Um levantamento só pode ser efetuado se existir saldo suficiente no balanço do cliente. Para cada levantamento e depósito é registado um novo movimento com a sua data respetiva (data corrente do sistema na hora do registo), o valor do balanço deve ser atualizado de acordo com a transação a ser efetuada.

    a. Crie um predicado para efetuar um depósito em determinado cliente.
    b. Crie um predicado para efetuar um levantamento em determinado cliente.
    c. Crie um método na classe Cliente que permita realizar um depósito de um determinado valor.
    d. Crie um método na classe Cliente que permita realizar um levantamento de um determinado valor.
    e. Adicione ambas as opções ao menu de cliente já criado.

10. Verificação de elegibilidade a crédito e conceção de crédito. Um cliente é elegível a crédito se o seu saldo no balanço total for superior a 100 unidades monetárias e o seu crédito for 0 ou inexistente. Quando o crédito é concedido o valor do balanço total é atualizado pela soma do balaço com o valor do crédito.
    a. Crie um predicado em Prolog que verifique se um determinado cliente é elegível a crédito.
    b. Crie um predicado em Prolog que conceda crédito de um determinado valor a um determinado cliente.
    c. Crie um método na classe Cliente que use o Prolog e verifique se o cliente é elegível a crédito e imprima uma mensagem positiva ou negativa.
    d. Crie um método na classe Cliente que use o Prolog que conceda um crédito de um determinado valor ao cliente e imprima a mensagem de sucesso ou insucesso.
    e. Adicione a opção de verificar elegibilidade de crédito e de conceder crédito ao menu de cliente.

Notas:

- É livre de alterar os nomes das classes para o que mais lhe convém.
- Não precisa de criar mais nada além dos requisitos, aproveite o seu tempo apenas naquilo que é pedido.
- Para o ajudar na questão de obtenção da data atual deixo aqui um predicado:

```
getCurrentDate(Date) :-
  get_time(Timestamp),
  stamp_date_time(Timestamp, DateTime, 'UTC'),
  date_time_value(year, DateTime, Year),
  date_time_value(month, DateTime, Month),
  date_time_value(day, DateTime, Day),
  format(atom(Date), '~|~`0t~d~2+~|~`0t~d~2+~|~d', [Day, Month, Year]).
```

- No relatório é valorizado a explicação das opções tomadas, isto é, uma explicação com pequenos excertos de código de como tomou as suas decisões, com um debate entre prós e contras. Apenas considere as decisões relevantes não necessita de descrever o seu código todo. Apresente alguns exemplos de funcionamento do seu programa nas mais variadas funcionalidades.