

Número: _____ Nome: _____ Turma: _____

Responda a cada Grupo em folhas separadas e identificadas com o seu número e nome.

Grupo I

Um clube desportivo, onde se praticam diversas modalidades, pretende uma aplicação informática para apoiar a gestão dos seus atletas e seccionistas.

Um seccionista é uma pessoa responsável pela gestão de uma modalidade, sendo caracterizado pelo nome, modalidade que gere e vencimento. Todos os seccionistas têm o mesmo vencimento.

O clube tem dois tipos de atletas: amadores e profissionais. Os atletas amadores são caracterizados pelo nome, número, profissão, modalidade que praticam e por não receberem qualquer pagamento do clube. Os atletas profissionais são caracterizados pelo nome, número, modalidade que praticam, nível de desempenho (ALTO, RAZOÁVEL ou BAIXO) e por auferirem um vencimento. Este vencimento tem uma parte fixa (vencimento base) e uma parte variável que depende do desempenho. Se o desempenho for ALTO é 50% do vencimento base, se for RAZOÁVEL é 10% do vencimento base e se for BAIXO é zero.

Assim:

1. Crie as classes adequadas para construir objetos representativos de atletas e seccionistas e que satisfaçam os seguintes requisitos:

- Cada classe tem de possuir apenas um construtor que deve ser o construtor completo (com parâmetros para inicializarem todos os campos de dados);
- Os métodos *get* e *set* NÃO precisam de ser especificados mas pode considerar que existem;
- O vencimento base dos atletas profissionais e o vencimento dos seccionistas devem poder ser alterados em qualquer momento pelo utilizador da aplicação;
- A obtenção dos vencimentos deve permitir uma programação genérica aplicável a qualquer entidade com vencimento;
- Cada atleta pratica só uma modalidade e cada seccionista é responsável por uma modalidade;
- As classes devem possibilitar o armazenamento de objetos em ficheiros binários;
- Interfaces e Enumerados devem ser utilizados sempre que tal seja adequado.

2. Crie uma classe principal, de nome *Main*, para testar as classes criadas anteriormente e que satisfaça os seguintes requisitos:

- Crie um objeto representativo de cada classe instanciável;
- Crie um contentor para guardar todos os objetos existentes;
- Guarde todos os objetos existentes no contentor;
- Imprima na consola uma listagem com a descrição de todos os objetos do contentor;
- Imprima na consola uma listagem com a descrição de todos os objetos com vencimento, apresentado para cada um deles o respetivo valor do vencimento.
- Armazene num ficheiro binário o contentor de objetos. Deve tratar adequadamente possíveis exceções.



Grupo II

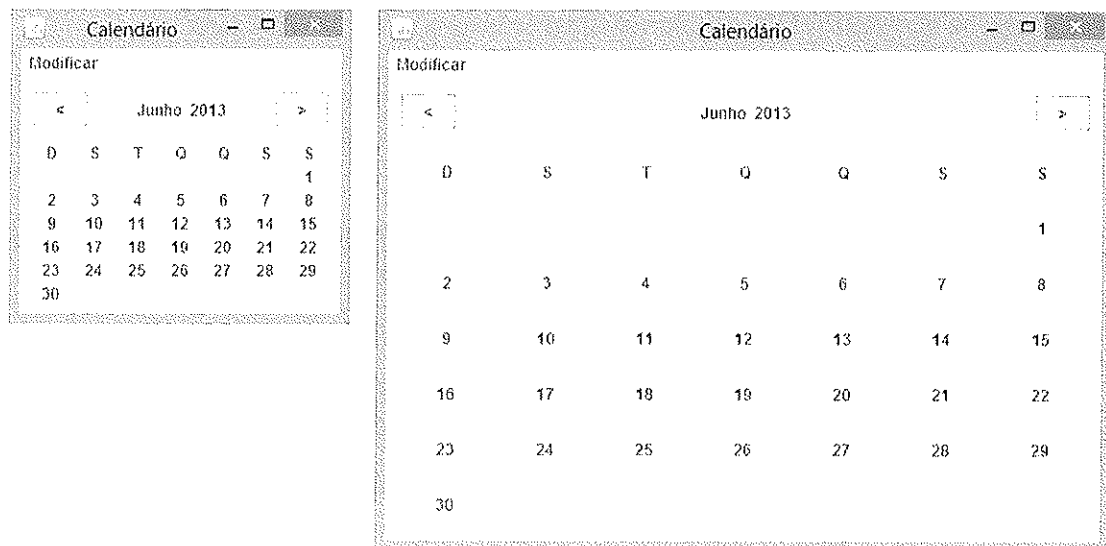
Pretende-se uma aplicação para mostrar um calendário, apresentando um mês de cada vez, conforme é ilustrado nas figuras seguintes. O mês e o ano iniciais são os atuais.



Os botões de comando < e > devem permitir retroceder ou avançar um mês, respetivamente.

Entre os botões de comando deve ser indicado o mês visualizado. O item *Ano* do menu *Modificar* deve mostrar a caixa de diálogo *Modificar o Ano*, para permitir ao utilizador da aplicação alterar o ano do mês apresentado.

O comportamento da janela durante o seu redimensionamento deve ser igual ao da figura seguinte.



Assim:

1. Desenhe um esquema com o posicionamento escolhido para os diferentes componentes gráficos, para que a janela tenha o comportamento ilustrado na figura anterior. Indique o nome das variáveis que no seu programa atribuiu aos principais painéis, assim como, os correspondentes gestores de posicionamento.
2. Elabore o código da aplicação. Considere a existência de uma classe **Data** com os seguintes membros:

```
public Data(int ano, int mes, int dia)    // valores de mes: 1, 2, 3, ...
public int getAno()
public int getMes()
public int getDia()
public void setAno(int ano)
public void setMes(int mes)
public void setDia(int dia)
public int diaDaSemana()                //Domingo=0, Segunda=1, Terça=2, ...
public String nomeDoMes()
public int diasDoMes()
public int semanasDoMes()              // quantidade de semanas do mes
```