



**Prova de
Programação
Informática**



Enunciado | Módulo 2 – Linguagem de Programação sem Interface Gráfica (Java ou C#)

1. Crie um projecto com o nome ChallengeAS com a respectiva classe e considere o cenário de uma competição de programação onde existem Entidades que concorrem, participantes dessas entidades e provas com os seguintes detalhes:
 - Participante: pretende-se guardar o número de identificação pessoal, nome, data de nascimento e Entidade a que pertence.
 - Entidade: pretende-se guardar o nome da entidade, distrito a que pertence e a indicação se é uma entidade pública ou privada.
 - Prova: pretende-se guardar um identificador único (PROG_JAVA_3 - significa uma programação em java avançado, PROG_SQL_1 - significa uma programação básica em SQL e assim sucessivamente), a descrição textual completa, os participantes que concorreram e a classificação obtida de cada um deles.
 - Deverá criar todos os métodos seletores, modificadores e construtores que considerar necessários para uma boa implementação futura dos objetos supracitados.
 - Caso seja necessário utilizar listas de objetos é sugerida a utilização de ArrayList para uma fácil e correta manipulação dos objetos em questão.
2. Crie o método **getIdade()** no objeto Participante para que seja possível obter a idade do participante atualizada à data/hora do sistema. O método deverá retornar um número inteiro.
3. Crie o método **getVencedor()** no objeto Prova para que seja possível saber qual foi o Participante vencedor de uma determinada prova. O Participante vencedor é o Participante com maior número de pontos. Em caso de empate, retorna o primeiro Participante encontrado. O método deverá retornar um objeto Participante e não apenas uma propriedade ou característica do mesmo.
4. Crie o método **insereParticipante(Participante a)** no objeto Prova para que seja possível adicionar um novo Participante à lista já existente para essa prova. O retorno deste método deverá ser um número inteiro com o valor 1 (um) se a inserção foi concluída com sucesso ou 0 (zero) no caso contrário (por motivo de registo de duplicados).
5. Crie o método **getStats()** no objecto Prova para que seja possível ter as seguintes estatísticas sobre uma prova:

NOME DA PROVA: <<descrição textual da prova>> | VENCEDOR: <<nome do Participante vencedor>>
ANÁLISE POR DISTRITOS:
<<Distrito C>> | Média | Nº de participantes
<<Distrito F>> | Média | Nº de participantes
E assim sucessivamente para toda a lista de distritos.

A ordenação dos distritos deverá ser por ordem decrescente na média obtida. A média e indicação do número de participantes é sempre tendo em conta a Prova específica.



O retorno do método deverá ser uma String.

6. Crie o conceito de **Staff** sendo este um objeto abstrato que não poderá ser instanciado. Desta forma, todos os membros do Staff possuem um IDInterno do ChallengeAS, data de nascimento e um nome. Este objeto deverá ter os métodos seletores e modificadores necessários para uma correta implementação do mesmo.
7. Crie o conceito de **ChefeOficina** que é uma extensão do Staff. Sabe-se que cada Chefe de Oficina recebe um valor em euros para participar no ChallengeAS (valor único), tem uma data de inscrição associada numa Prova específica. Este objeto deverá ter os construtores, métodos seletores e modificadores necessários para uma correta implementação do mesmo.
8. Crie o método **calculaDataFimParticipacao(int numeroDeDias)** no objeto ChefeOficina para que seja possível ter como output a data de término da participação de um Chefe de Oficina com base no argumento entregue (numeroDeDias). Desta forma, o seu método deverá conseguir somar os dias recebidos à data de inscrição. Não é possível ter datas como: 34 de Maio, 30 de Fevereiro, etc. O retorno do método deverá ser uma String.
9. Crie o método **calculaRetencao()** no objeto ChefeOficina para que o seja possível calcular automaticamente qual a retenção a aplicar sabendo que todos os Chefes de Oficina possuem uma taxa de retenção de 25%. O retorno do método deverá ser um número decimal.
10. Na classe com main, crie uma estrutura de dados para guardar a informação sobre todas as Provas, Participantes, Entidades e respetivos Chefes de Oficina. Note-se que a escolha de uma (ou várias) listas ou qualquer estrutura a seu gosto é da sua inteira responsabilidade e poderá a sua escolha simplificar (ou dificultar) futuras implementações de funcionalidades nesta prova. Nesta mesma classe, crie um menu, em modo consola, para que seja possível o utilizador ter várias opções e apenas sair da sua aplicação quando escolher a opção 0 - Sair.
11. Adicione ao menu anterior a opção 1 - Adicionar novo Participante em que o objetivo é adicionar à sua estrutura de dados mais um objeto Participante preenchendo o utilizador todos os dados necessários para a criação do objeto deste tipo. Não será possível ter dois participantes com o mesmo número de identificação pessoal. Crie todos os métodos adicionais necessários nos objetos que considerar necessários para uma correta implementação deste ponto.
12. Adicione ao menu anterior a opção 2 - Adicionar nova Entidade em que o objetivo é adicionar à sua estrutura de dados mais um objeto Entidade preenchendo o utilizador todos os dados necessários para a criação do objeto deste tipo. Não será possível ter duas entidades com o mesmo nome no mesmo distrito (mas podem ter o mesmo nome desde que distritos diferentes). Crie todos os métodos adicionais necessários nos objetos que considerar necessários para uma correta implementação deste ponto.



13. Adicione ao menu anterior a opção 3 - Adicionar Prova em que o objetivo é adicionar à sua estrutura de dados mais um objeto Prova sabendo que o utilizador começa por indicar o identificador único da prova (deverá indicar uma mensagem de erro se esse identificador ainda não existir) e logo de seguida indica quantos participantes pretende inserir (este número não pode ser igual ou inferior a 0 (zero)). Após essa indicação insere o número de identificação de cada participante (que necessita de estar já registado) e respetiva classificação e em caso de sucesso do primeiro participante, avança para o segundo e assim sucessivamente até ter inserido todos os participantes. Crie todos os métodos adicionais necessários nos objetos que considerar necessários para uma correta implementação deste ponto.
14. Adicione ao menu anterior a opção 4 - Gestão de erros em que o objetivo é remover um registo que tenha sido mal inserido (ou inserido por engano). Assim, o utilizador deverá indicar se pretende remover um Participante, uma Entidade ou uma inscrição numa Prova. Deverá fornecer o número de identificação pessoal para remover o participante ou a sua inscrição ou o nome da entidade e o distrito no caso da remoção de uma Entidade. Crie todos os métodos adicionais necessários nos objetos que considerar necessários para uma correta implementação deste ponto.
15. Adicione ao menu anterior a opção 5 - Adicionar Chefe de Oficina em que o objetivo é adicionar à sua estrutura de dados mais um objeto ChefeOficina sabendo que o utilizador indica todos os dados necessários. O registo só fica concluído com sucesso se todas as informações forem inseridas (e validadas - Prova tem de existir / Só pode existir um Chefe de Oficina por Prova). Crie todos os métodos adicionais necessários nos objetos que considerar necessários para uma correta implementação deste ponto.
16. Adicione ao menu anterior a opção 6 - Stats ChallengeAS - Provas em que o objetivo é receber alguns dados estatísticos sobre a competição, mais concretamente:

> Para cada Prova mostrar a análise de distritos (usar métodos já criados anteriormente)
> Qual a competição com maior número de participantes? Deverá indicar a descrição textual da Prova e o respetivo número.

Segue um exemplo do output esperado:

Stats ChallengeAS - Provas

NOME DA PROVA: <<descrição textual da prova>> | VENCEDOR: <<nome do Participante vencedor>>

ANÁLISE POR DISTRITOS:

<<Distrito C>> | Média | Nº de participantes

<<Distrito F>> | Média | Nº de participantes

....

MENOR NÚMERO DE PARTICIPANTES:

<<Nome da Prova>> | <<Número de Participantes>>



XXII CAMPEONATO REGIONAL AÇORES 2017

17. Adicione ao menu anterior a opção 7 - Stats ChallengeAS - Participantes em que o objetivo é receber alguns dados estatísticos sobre a competição, mais concretamente:

> Qual o nome do participante com pior nota no campeonato (independentemente da Prova)? Deverá indicar o nome do Participante, a descrição textual da Prova e a respetiva classificação obtida.

Segue um exemplo do output esperado:

Stats ChallengeAS - Participantes ##### PIOR NOTA:

<<Nome do Participante>> | <<Nome da Prova>> | <<Classificação obtida>>

18. Adicione ao menu anterior a opção 8 - Stats ChallengeAS - ChefesOficina em que o objetivo é receber alguns dados estatísticos sobre a competição, mais concretamente:

> Qual o TOP3 das Provas mais vantajosas financeiramente? Sabe-se que a prova mais vantajosa financeiramente é a prova que se paga menor valor ao Chefe de Oficina.

A ordenação do TOP3 deverá ser por ordem crescente de valores pagos aos chefes de oficina.

Para facilitar a leitura deverá ainda realizar um pictograma como é mostrado no seguinte exemplo do output esperado:

Stats ChallengeAS - ChefesOficina ##### TOP3 FINANCEIRO (LOW BUDGET)

LUGAR	PROVA	CUSTO	PICTOGRAMA
1	<<nome da prova>>	10 (exemplo)	€€€€€€€€€€
2	<<nome da prova>>	14 (exemplo)	€€€€€€€€€€€€
3	<<nome da prova>>	20 (exemplo)	€€€€€€€€€€€€€€€€



19. Adicione ao menu anterior a opção 9 - Stats ChallengeAS - Finanças em que o objetivo é mostrar ao utilizador quanto é que o ChallengeAS rendeu de impostos para o Estado Português sabendo que o utilizador deverá ver o valor final, em euros, mas também os totais parciais. Entram para os cálculos os seguintes itens:

> Retenção dos Chefes de Oficina

> Vencedor de cada Prova tem de pagar um montante com base no número de participantes da mesma, isto é, o vencedor de uma determinada prova paga:

0 (zero) euros se a prova tiver no máximo dois participantes. 100 euros se a prova tiver entre 3 e 5 participantes.

250 euros se a prova tiver entre 6 e 10 participantes. 500 euros em provas com mais de 10 participantes.

Segue um exemplo do output esperado:

Stats ChallengeAZ - Finanças ##### RETENÇÃO DE CHEFES DE
OFICINA: <<valor>> EUR

VENCEDOR DA PROVA <<id da prova1>>: <<valor1>> EUR VENCEDOR DA PROVA <<id da prova2>>:
<<valor2>> EUR

...

TOTAL A PAGAR AO ESTADO PORTUGUÊS RELATIVO A IMPOSTOS: <<valor>> +
<<valor1>> + <<valor2>> + ... EUR