

Questão 1

Escreva uma classe Aluno, cujos objetos representam alunos matriculados em uma disciplina. Cada objeto dessa classe deve guardar os seguintes dados do aluno: matrícula, nome, duas notas de prova e uma nota de trabalho. Escreva os seguintes métodos para esta classe:

construtor	define os valores iniciais dos atributos da classe
Getters...	Métodos para retornar valores de cada atributo
Setters...	Métodos para alterar valores de cada atributo
toString	Retorna os valores dos atributos do objeto como uma String
media	calcula a média final do aluno (cada prova tem peso 2,5 e o trabalho tem peso 2)
final	calcula quanto o aluno precisa para a prova final (retorna zero se ele não for para a final)

Questão 2

Escreva uma classe Data cuja instância (objeto) represente uma data. Esta classe deverá dispor dos seguintes métodos:

construtor	define a data que determinado objeto (através de parâmetro), este método verifica se a data está correta, caso não esteja a data é configurada como 01/01/0001
Getters...	Métodos para retornar valores de cada atributo
Setters...	Métodos para alterar valores de cada atributo
toString	Retorna os valores dos atributos do objeto como uma String
compara	recebe como parâmetro um outro objeto da Classe data, compare com a data corrente e retorne: 0 se as datas forem iguais; 1 se a data corrente for maior que a do parâmetro; -1 se a data do parâmetro for maior que a corrente.
getMesExtenso	retorna o mês da data corrente por extenso
isBissexto	retorna verdadeiro se o ano da data corrente for bissexto e falso caso contrário

Questão 3

Escreva uma classe em que cada objeto representa um voo que acontece em determinada data e em determinado horário. Cada voo possui no máximo 100 passageiros, e a classe permite controlar a ocupação das vagas. A classe deve ter os seguintes métodos:

construtor	configura os dados do voo (recebidos como parâmetro): número do voo;
Getters...	Métodos para retornar valores de cada atributo
Setters...	Métodos para alterar valores de cada atributo
toString	Retorna os valores dos atributos do objeto como uma String
verifica	verifica se o número da cadeira recebido como parâmetro está ocupada

ocupa	ocupa determinada cadeira do voo, cujo número é recebido como parâmetro, e retorna verdadeiro se a cadeira ainda não estiver ocupada (operação foi bem sucedida) e falso caso contrário
vagas	retorna o número de cadeiras vagas disponíveis (não ocupadas) no voo

Questão 4

Implemente a classe *Questao*, para representar questões de múltipla escolha. Cada questão deve ter um número inteiro identificador, um texto (String) de enunciado da questão; uma lista de textos (String) para as alternativas e o gabarito da questão, indicando a alternativa correta.

A classe deve ter os seguintes métodos implementados:

construtor	Deve receber o número da questão, o texto do enunciado, a lista de alternativas e o gabarito.
Getters...	Métodos para retornar valores de cada atributo
Setters...	Métodos para alterar valores de cada atributo
toString	Retorna os valores dos atributos do objeto como uma String

Questão 5

Escreva uma classe *Prova* em que cada objeto representa uma prova feita por um aluno. Esta prova possui questões de múltipla escolha (letras A a E) criados a partir de objetos *Questao* do exercício anterior. A prova tem valor total de 10,0 pontos. O valor de cada questão é o mesmo e depende da quantidade de questões inseridas na prova. Esta classe deverá controlar as questões respondidas pelo aluno. Para isto, a classe deve implementar os métodos:

construtor	recebe como parâmetro uma lista de objetos da classe <i>Questao</i>
respostaAluno	recebe como parâmetro a resposta dada pelo aluno a uma questão
acertos	retorna a quantidade de questões que o aluno acertou
erros	retorna a quantidade de questões que o aluno errou
nota	retorna a nota que o aluno tirou na prova

Questão 6

A maioria das linguagens de programação dispõem de um suporte nativo a vetores (arrays), que exigem a definição de seu tamanho no momento da sua criação. Depois de criado, o tamanho do vetor não pode ser modificado. Escreva uma classe chamada *Vetor* cujos objetos simulem vetores de tamanho variável. A classe deve definir os seguintes métodos:

construtor	recebe como parâmetro o tamanho inicial do vetor
insert	recebe como parâmetro um valor do tipo Integer e a coloca na próxima posição disponível do vetor; note que o vetor cresce automaticamente, portanto, se a inserção ultrapassar o tamanho inicial estabelecido na criação, por exemplo, o vetor deve aumentar seu tamanho automaticamente

get	recebe como parâmetro uma posição do vetor e retorna o valor Integer que estiver naquela posição; se a posição não estiver ocupada ou ultrapassar o tamanho do vetor, este método retorna nulo (não deve lançar exceção)
size	retorna o número de elementos inseridos no vetor (vetor vazio retorna zero)

Questão 7

Crie uma classe OrdenadorArray que tem como métodos:

ordenarCrescente	recebe um vetor de Integers e ordena esses valores em ordem crescente
ordenarDecrescente	recebe um vetor de Integers e ordena esses valores em ordem decrescente

Questão 8

Crie uma classe VetorOrdenado que é uma variante da classe anterior, com as seguintes modificações e acréscimos:

insert	recebe um valor Integer como parâmetro e a insere em uma posição tal que o vetor de valores semantenha sempre em ordem crescente
merge	recebe como parâmetro um outro objeto da classe VetorOrdenado e retorna um terceiro objeto da classe VetorOrdenado (criado dentro do método) que contém a mescla (merge) do vetor corrente com aquele recebido como parâmetro

A classe VetorOrdenado deve usar um objeto OrdenadorArray do exercício anterior para ser implementada.