

Lista de Questões 5
Programação Orientada a Objetos
INSTITUTO FEDERAL DE ALAGOAS - CAMPUS RIO LARGO

1. Crie uma classe Agenda que pode armazenar 10 pessoas e que seja capaz de realizar as seguintes operações:

armazenarPessoa(String nome, int idade, float altura);	
removerPessoa(String nome)	
buscaPessoa(String nome)	informa em que posição da agenda está a pessoa
imprimirAgenda	imprime os dados de todas as pessoas da agenda
imprimirPessoa(int index)	imprime os dados da pessoa que está na posição “i” da agenda.

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
compararDatas	recebe como parâmetro um outro objeto da Classe data, compare com a data corrente e retorne: <ul style="list-style-type: none">• 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
Getters & Setters	Metódos get e set que se fizerem necessarios.

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, data (para armazenar a data utilize um objeto da classe Data, criada na questão anterior);
proximoLivre	retorna o número da próxima cadeira livre
verificarOcupacao	verifica se o número da cadeira recebido como parâmetro está ocupada
ocupar	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
quantidadeVagas	retorna o número de cadeiras vagas disponíveis (não ocupadas) no voo
Getters & Setters	Metódos get e set que se fizerem necessarios.

4. Considere a seguinte classe, cujo método `respostaQuestao` recebe como parâmetro o número de uma questão e retorna a sua resposta correta, proveniente de um gabarito.

```
public class Gabarito {
    public char respostaQuestao(int numeroQuestao) {

    }
}
```

Escreva uma classe `Prova` em que cada objeto representa uma prova feita por um aluno. Esta prova possui 15 questões de múltipla escolha (letras A a E). As 10 primeiras questões valem 0,5 ponto e as 5 últimas questões valem 1 ponto. Esta classe deverá controlar as questões respondidas pelo aluno. Para isto, a classe deve implementar os métodos:

construtor	recebe como parâmetro um objeto da classe <code>Gabarito</code> contendo o gabarito da prova.
<code>respostaAluno</code>	recebe como parâmetro a resposta dada pelo aluno a uma questão; este método não recebe entre os parâmetros o número da questão, ele

	mesmo deve estabelecer um controle interno de modo que as questões sejam inseridas sequencialmente, ou seja, a primeira vez que o método é chamado, insere a primeira questão, a segunda, insere a segunda questão, e assim por diante.
acertos	retorna a quantidade de questões que o aluno acertou.
nota	retorna a nota que o aluno tirou na prova.
maior	recebe como parâmetro um outro objeto da classe Prova e retorna a nota do aluno que acertou mais questões; se houver empate, retorna a maior nota; se houver empate novamente, retorna a nota do aluno representado no objeto corrente.

5. Identifique as classes e implemente um programa para a seguinte especificação: “O supermercado vende diferentes tipos de produtos. Cada produto tem um preço e uma quantidade em estoque. Um pedido de um cliente é composto de itens, onde cada item específico o produto que o cliente deseja e a respectiva quantidade. Esse pedido pode ser pago em dinheiro, cheque ou cartão.”

6. Faça um programa para representar a árvore genealógica de uma família. Para tal, crie uma classe Pessoa que permita indicar, além de nome e idade, o pai e a mãe. Tenha em mente que pai e mãe também são do tipo Pessoa.

7. Faça um programa que calcule a área de uma figura geométrica. Aceite quatro tipos de figura geométrica: quadrado, retângulo, triângulo e círculo. Use herança e polimorfismo.

8. Crie uma classe Animal que obedeça à seguinte descrição:
 - ❖ Possua os atributos
 - nome (String);
 - comprimento (float);
 - número de patas (int);
 - cor (String);
 - ambiente (String);
 - velocidade média (float) .

- ❖ Crie um método construtor que receba por parâmetro os valores iniciais de cada um dos atributos e atribua-os aos seus respectivos atributos.
- ❖ Crie os métodos get e set para cada um dos atributos.
- ❖ Crie um método dados, sem parâmetro e do tipo void, que, quando chamado, imprime na tela uma espécie de relatório informando os dados do animal.

Crie uma classe Peixe que herde da classe Animal e obedeça à seguinte descrição:

- ❖ Possua um atributo - característica(String)
- ❖ Crie um método construtor que receba por parâmetro os valores iniciais de cada um dos atributos (incluindo os atributos da classe Animal) e atribua-os aos seus respectivos atributos.
- ❖ Crie ainda os métodos get e set para o atributo característica.
- ❖ Crie um método dadosPeixe sem parâmetro e do tipo void, que, quando chamado, imprime na tela uma espécie de relatório informando os dados do peixe (incluindo os dados do Animal e mais a característica).

Crie uma classe Mamifero que herde da classe Animal e obedeça à seguinte descrição:

- ❖ possua um atributo alimento(String)
- ❖ Crie um método construtor que receba por parâmetro os valores iniciais de cada um dos atributos (incluindo os atributos da classe Animal) e atribua-os aos seus respectivos atributos.
- ❖ Crie ainda os métodos get e set para o atributo alimento.
- ❖ Crie um método dadosMamifero sem parâmetro e do tipo void, que, quando chamado, imprime na tela uma espécie de relatório informando os dados do mamifero (incluindo os dados do Animal e mais o alimento).

Crie uma classe TestarAnimais possua um método main para testar as classes criadas.

- ❖ Crie um objeto camelo do tipo Mamífero e atribua os seguintes valores para seus atributos:
 - Nome: Camelo
 - Comprimento: 150 cm
 - Patas: 4
 - Cor: Amarelo
 - Ambiente: Terra
 - Velocidade: 2.0 m/s
- ❖ Crie um objeto tubarao do tipo Peixe e atribua os seguintes valores para seus atributos:

- Nome: Tubarão
- Comprimento: 300 cm
- Patas: 0
- Cor: Cinzento
- Ambiente: Mar
- Velocidade: 1.5 m/s
- Característica: Barbatanas e cauda

❖ Crie um objeto ursocanada do tipo Mamifero e atribua os seguintes valores para seus atributos:

- Nome: Urso-do-canadá
- Comprimento: 180 cm
- Patas: 4
- Cor: Vermelho
- Ambiente: Terra
- Velocidade: 0.5 m/s
- Alimento: Mel

Chame os métodos para imprimir os dados de cada um dos objetos criados.