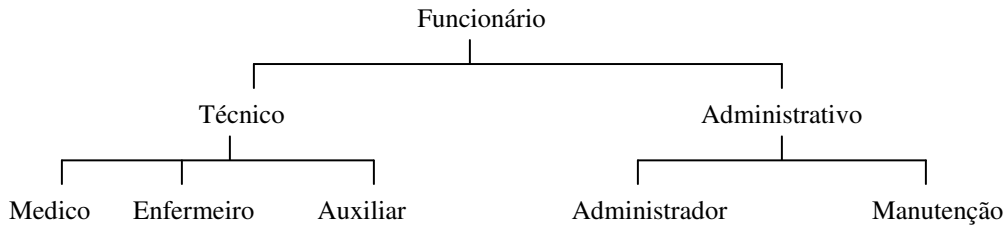


## Segunda Lista de Exercícios de Programação III

Um Hospital de Vitória te contratou para desenvolver um protótipo de um sistema de informação que cadastre os funcionários e salas do hospital e gere alguns relatórios.

1. Considere que durante a especificação do sistema foram identificadas as seguintes categorias de funcionários classificadas na hierarquia abaixo:



Considere também que:

- Todo funcionário possui um código identificador único (um número inteiro), um nome (uma string), um salário (um número de ponto flutuante) e uma data de contratação.
- Todo técnico possui um atributo representando o número de anos que exerce sua atividade profissional (um número inteiro).
- Todo médico possui uma especialidade (uma string) e um identificador único (um número inteiro) no Conselho Regional de Medicina (CRM). São exemplos de especialidades: “otorrinolaringologia”, “pediatria”, “cirurgia geral”, etc.

Com base nessas informações, implemente as classes *Funcionario*, *Tecnico* e *Medico*. Implemente também uma classe *Data* com atributos específicos para dia, mês e ano. Garanta que todas subclasses concretas de *Funcionario* implementem obrigatoriamente os métodos de leitura de dados da entrada padrão e de impressão dos dados na saída padrão.

2. Implemente uma classe *Sala* que possui como atributos o seu número (um número inteiro), o código identificador do funcionário responsável e sua disponibilidade (um booleano).
3. Implemente uma classe *Lista* genérica na qual os elementos da lista são armazenados em uma lista encadeada dos elementos.
4. Considerando que todas as classes da hierarquia foram implementadas (as que você não implementou na primeira questão foram implementadas por outro programador), utilize-as juntamente com a classe *Lista* para implementar um programa que:
  - leia da entrada padrão o número total de funcionários do hospital
  - leia da entrada padrão os dados de cada funcionário do hospital
  - leia da entrada padrão o número total de salas
  - leia da entrada padrão os dados de cada sala do hospital
  - imprima os dados de todas as salas do hospital ordenadas crescentemente por seu número
  - imprima os dados de todos os médicos do hospital cujo especialidade é “cirurgia plastica” e que foram contratados nos últimos três anos a partir da data de hoje.
  - imprima os dados de todas as salas ocupadas do hospital que têm médicos como funcionário responsável
  - imprima os dados de todos os funcionários marajás do hospital ordenados decrescentemente por salário (um marajá tem salário superior a 10.000 reais)
  - imprima os dados de todos os médicos pediatras ordenados crescentemente por anos de atividade
5. Uma firma de Engenharia te contratou para desenvolver um protótipo em Java de um sistema de desenho auxiliado por computador no plano cartesiano bidimensional. Para implementar esse protótipo você deve atentar para os aspectos de uma boa programação orientada a objetos (em especial, ao ocultamento de informação) e deve:

6. Criar uma classe Ponto representada por duas coordenadas inteiras.
7. Criar uma classe abstrata FormaGeometrica, a qual contém um ponto em sua origem. A classe FormaGeometrica deve possuir pelo menos um método abstrato para impressão dos dados de uma forma geométrica.
8. Criar uma classe concreta Linha, subclasse de FormaGeometrica. Além de um ponto de origem, a classe Linha tem um outro ponto representando sua outra extremidade. Um método específico dessa classe que deve ser implementado é aquele que calcula o tamanho da linha. O tamanho da linha é dado pela seguinte fórmula

$$d = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}$$

9. Criar uma classe concreta Circulo, subclasse de FormaGeometrica. Além de sua origem, a classe Circulo possui um raio e uma área. Um método específico dessa classe que deve ser implementado é aquele que calcula a área do círculo. A área do círculo é dada pela fórmula

$$a = \pi R^2$$

Você pode considerar o valor 3.1416 para o número  $\pi$ .

10. Criar uma classe concreta Triangulo, subclasse de FormaGeometrica. Além de um ponto de origem representando um vértice do triângulo, esta classe possui ainda dois outros pontos representando os seus outros vértices. Essa classe deve possuir um método que responda se um triângulo é equilátero (todos os seus lados são iguais).
11. Criar uma classe Desenho, a qual pode conter um número indefinido de formas geométricas (isto é, linhas, círculos e triângulos).

Você deve implementar os seguintes métodos dessa classe:

- a) Ordenar as formas pertencentes ao desenho segundo algum critério de ordenação a ser especificado na hora da chamada do método.
- b) Imprimir os dados de todas as formas do desenho ordenados pela coordenada x da origem da forma geométrica (voce deve usar o método definido em a).
- c) Imprimir os dados de todos os círculos de área superior a 20 (com exceção daqueles com raio negativo ou zero, para os quais, em vez dos dados, deve ser impressa uma mensagem indicativa dessa anomalia) ordenados crescentemente pelo sua área (voce deve usar o método definido em a)

12. Leia o seguinte trecho de programa em C:

```

int leArquivo ( int v [ ] ) {
    char *n;
    int cod;
    FILE *p;
    cod = leNomeArq (n);
    if (cod == 0) {
        printf ("nome invalido");
        return -1;
    }
    cod = abreArq (n, p);
    if (cod == 0) {
        printf (
            "arquivo inexistente");
        return -1;
    }
    cod = carregaArq (p, v, 100);
    if (cod == 0) return -2;
    cod = fechaArq (p);
    if (cod == 0) return -3;
    return 0;
}

int tentaLer (int v [ ] ) {
    int cod;
    do {
        cod = leArquivo (v);
        if (cod == -1) {
            if (!continua ( ))
                return cod;
        } else {
            return cod;
        }
    } while (1);
}

main ( ) {
    int cod;
    int vet [100];
    cod = tentaLer (vet);
    switch (cod) {
        case 0: break;
        case -1:
            printf ("erro de nome");
            break;
        case -2:
            printf ("erro de carga");
            break;
        case -3:
            printf ("erro de fechamento");
    };
    ordena (vet);
    imprime (vet);
}

```

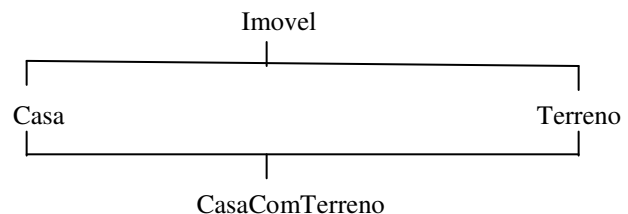
Considere que as funções *leNomeArq*, *abreArq*, *carregaArq* e *fechaArq* retornam 0 (zero) se não forem bem sucedidas e 1 (um), caso contrário. Considere também que a função *continua* pergunta ao usuário se ele deseja tentar novamente e retorna 1 (um) em caso afirmativo e 0 (zero) em caso negativo. Refaça esse programa usando classes e o mecanismo de tratamento de exceções de Java. Considere que na versão de Java, as funções *leNomeArq*, *abreArq*, *carregaArq* e *fechaArq* retornam *void* mas disparam respectivamente as seguintes exceções *nomeExc*, *arqExc*, *cargaExc* e *fechaExc*.

13. Considere o seguinte trecho de código em JAVA. Mostre o que será impresso nas chamadas *Teste.f(1)*, *Teste.f(2)* e *Teste.f(3)*.

```
public class Teste {
    public static void f(int i) {
        try {
            switch (i) {
                default:
                    case 1: throw new Exception();
                    case 2: throw new NumberFormatException();
                    case 3: throw new EOFException();
            }
        } catch (Exception e) { System.out.println("Exception");}
        catch (NumberFormatException e) { System.out.println("NumberFormatException");}
        catch (EOFException e) { System.out.println("EOFException");}
    }
}
```

14. Uma Imobiliária de Vitória te contratou para desenvolver um protótipo de um sistema de informação para cadastrar os imóveis de sua carteira de ofertas. Considere que durante a especificação do sistema foram identificadas as seguintes categorias de imóveis classificadas na hierarquia abaixo. Considere também que:

- Todo imóvel possui um preço (um número ponto flutuante) e um identificador (um número inteiro).
- Todo casa possui um número de quartos (um número inteiro).
- Toda terreno possui uma área (um número ponto flutuante).



Com base nessas informações, implemente em JAVA as classes *Imovel*, *Casa*, *Terreno* e *CasaComTerreno*. **Assegure** que toda subclasse concreta de *Imovel* tenha de implementar os métodos de leitura (do console) e impressão dos seus dados (no console). Implemente um método na classe *Casa* que identifique se ela tem um determinado número mínimo de quartos. Implemente um método na classe *Terreno* que identifique se ele tem um determinada área mínima. Faça um programa que leia um número de imóveis e, para cada imóvel, leia o tipo do imóvel, crie um objeto do imóvel específico, leia os dados do objeto e o inclua em uma estrutura de dados (pode ser um vetor ou uma coleção da biblioteca padrão de Java). Após o final da leitura, imprima uma listagem com os dados de cada casa com mais de quatro quartos. Repita o procedimento para imprimir uma listagem com os dados de cada terreno com área superior a 1000 metros quadrados.

15. Crie uma GUI que simula o comportamento de um telefone celular simples. Nesta GUI, além das teclas de dígitos, ligar e desligar, deve ter um visor indicativo do número discado. Faça com que toda vez que seja feita uma ligação seja apresentada na saída padrão o número do telefone discado. Faça também que o tempo da ligação seja mostrado no visor e na saída padrão quando a ligação for encerrada.