



# Tecnologia em Análise e Desenvolvimento de Sistemas

## Programação Orientada a Objetos Caderno de Exercícios



1º/2018





# Introdução a Programação em Java



# Exercício 1

- Utilize a internet para descobrir a população mundial atual e a taxa de crescimento demográfico mundial anual. Escreva um aplicativo que introduza esses valores e, então, que exiba a população mundial estimada depois de um, dois, três, quatro e cinco anos.
- Fonte: Deitel e Deitel. Java Como Programar, ed. 10, Exercício 2.34



## Exercício 2

- Pesquise sobre faixa solidária. Crie um aplicativo que calcule o custo diário de dirigir, para estimar quanto dinheiro pode ser economizado com o uso da faixa solidária, que também tem outras vantagens, como reduzir emissões de carbono e congestionamento de tráfego. O aplicativo deve introduzir as seguintes informações e exibir o custo por dia de dirigir para o trabalho do usuário:
  - Quilômetros totais dirigidos por dia.
  - Preço por litro de gasolina.
  - Quilômetros médios por litro.
  - Taxas de estacionamento por dia.
  - Pedágio por dia.
- Fonte: Deitel e Deitel. Java Como Programar, ed. 10, Exercício 2.35



## Exercício 3

- Implemente um sistema orientado a objetos que armazene informações de cinco alunos.
- Para cada aluno sabe-se que os dados são prontuário, nome, nota prova 1, nota prova 2, nota dos exercícios e nota do projeto. Cada nota possui um peso idêntico ao da disciplina POOS3.
- O sistema deve informar:
  - Quantidade de alunos aprovados;
  - Quantidade de alunos reprovados;
  - Média geral da turma.
- Utilize na implementação JoptionPane para entrada e saída de dados.



## Exercício 4

- Implemente um sistema para uma calculadora de inteiros.
- A calculadora possui, além do construtor, os métodos:
  - `private int soma(int n);`
  - `private int subtrai(int n);`
  - `private int multiplica(int n);`
  - `private int divisao(int n);`
  - `public int calcular(int operando, char operador);`
  - `public void limparMemoria();`
- Implemente vários casos de testes para utilizar a calculadora.



# Exercício 5

- Implemente a classe caneta:
  - Atributos
    - cor
    - carga
    - tampa
    - tipo
  - Método
    - escreverPalavra()
  - Observação:
    - Para cada palavra escrita uma quantidade da carga é gasta.
- Implemente também um programa principal para executar as operações de uma caneta.



# Exercícios 6

- Um colecionador de figurinhas deseja que um sistema computadorizado controle sua coleção.
  - O álbum de figurinhas possui 150 figurinhas diversas, numeradas de 1 a 150.
  - O colecionador pode adquirir figurinhas, cada pacote possui três figurinhas. Não sabe-se qual figurinha está no pacote ou mesmo se existe figurinhas repetidas no pacote.
- Implemente um sistema que controle o álbum de figurinhas do colecionador e também o permita o controle de figurinhas repetidas.
- O sistema deve permitir a aquisição de figurinhas, pacote com 3 figurinhas. O pacote é formado por figurinhas geradas aleatoriamente.
- O sistema deve informar quantas figurinhas o álbum precisa para ficar completo e quais são essas.
- O sistema deve informar também quais as figurinhas repetidas e a quantidade de cada uma das repetidas que o colecionador possui. Lembre-se que uma figurinha está no álbum, somente as adquiridas adicionais igual a do álbum são repetidas.
- Implemente um sistema orientado a objetos para esse gerenciamento.





# Exercício 7

- Considere um baralho de 52 cartas.
- Cada carta possui a indicação do naipe e do valor.
- Implemente um sistema que gere as 52 cartas do baralho.
- Utilize um algoritmo do tipo INSERTSORT para ordenar as cartas pelos naipes (paus, copas, espadas, ouros) e valores (A, 2,3,4,5,6,7,8,9,10,J,Q,K).
- Implemente um algoritmo que embaralhe o baralho.
- Implemente um sistema para testar as implementações realizadas.



## Exercício 8

- Considerando o Projeto “Fixacao8” disponibilizado no github, implemente adequadamente o classe MinhaFila.java.
- Implemente um programa principal para interagir com a MinhaFila implementada.
  - A fila pode ser baseada em array, array circular ou listas encadeadas.

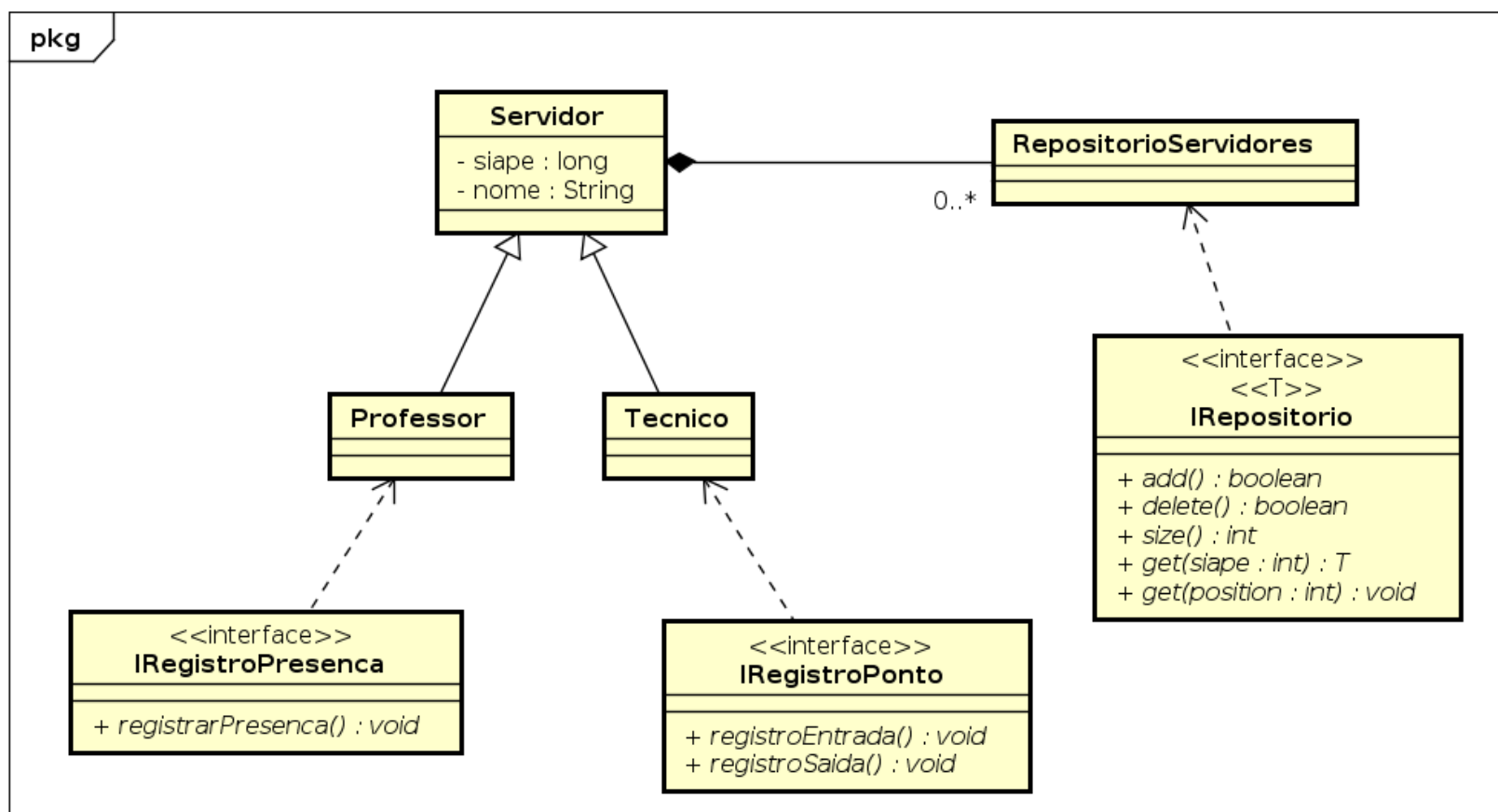


# Exercício 9

- Implemente um sistema de controle de acesso.
  - O sistema deve permitir a entrada de pessoas em um ambiente após a verificação das credenciais da pessoa. Deve-se observar que cada ambiente possui uma especificidade, nome e uma senha pessoal.
  - Deve existir um único array que armazene todas as pessoas em qualquer ambiente.
  - O sistema será implantado em:
    - Escola
      - Aluno possui registro de matrícula (Ex.: 180001)
      - Professor possui prontuário (Ex.: er1234)
    - Hospital
      - Paciente possui número da ficha (Ex.: P1234)
      - Médico possui CRM (Ex.: 123-x)
      - Visitante possui código (Ex.: 1234P1234) e está vinculado a um paciente
    - Área VIP

# Exercícios 10

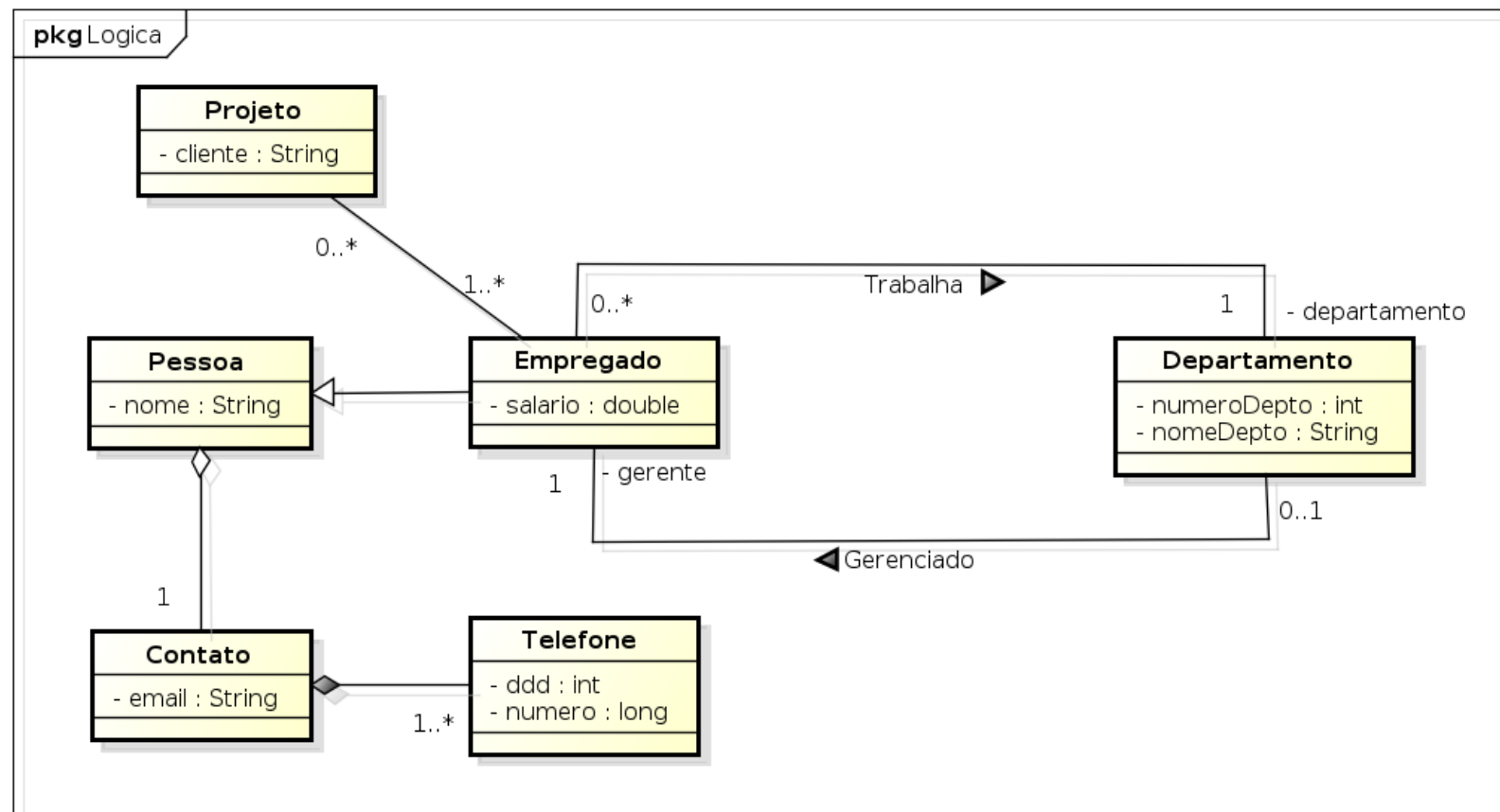
- Implementar o sistema de cadastro de servidores do IFSP seguindo o modelo.





# Exercício 11

- Implemente um sistema que resolva o problema apresentado no diagrama de classes:





# Exercício 12

- Implemente um sistema orientado a objetos para gerenciar as matrículas de alunos de uma escola.
- A escola oferta vários cursos, sobre cada curso deve-se armazenar: nome do curso, carga horária e número de vagas.
- A escola aceita a inscrição de vários alunos, sobre cada aluno deve-se armazenar: nome, número de matrícula e e-mail.
- Sobre as matrículas, um aluno pode se matricular, uma única vez, em cada curso oferecido pela escola. Nos cursos podem se matricular vários alunos, o máximo é definido pelo número de vagas.
- Além do cadastro e matrícula, o sistema deve emitir dois relatórios:
  - Alunos matriculados em um curso;
  - Cursos em que um aluno foi matriculado;