

## Lista de Exercícios 1 - POO-II - INE5404

**Warm up** (\*Para a entrega destes exercícios, coloque todas as soluções em um mesmo arquivo, separando os trechos de códigos por comentários que indicam à qual warm up o código está relacionado.)

- 1) Crie a classe Televisao com os atributos *ligada* (inicializado com valor False) e *canal* (inicializado com valor 2).
- 2) Adicione os atributos tamanho e marca à classe Televisao. Crie dois objetos Televisao e atribua tamanhos e marcas diferentes. Depois, imprima o valor desses atributos de forma a confirmar independência dos valores de cada instância (objeto).
- 3) Adicione dois novos métodos muda\_canal\_para\_cima e muda\_canal\_para\_baixo. Atualmente, a classe Televisao inicializa o canal com 2. Modifique a classe televisao de forma a receber o canal inicial em seu construtor.
- 4) Adicione mais dois atributos canal\_minimo (valor padrão 1) e *canal\_maximo* (valor padrão 99) e modifique a classe Televisao de forma que, se pedirmos para mudar o canal para baixo, além do mínimo, ela vá para o canal máximo. Se mudarmos para cima, além do canal máximo, que volte ao canal mínimo.
- 5) Modifique o construtor da classe Televisao de forma que canal\_minimo e canal\_maximo seja parametros opcionais valendo respectivamente 2 e 14.
- 6) Crie duas instancias de Televisao, especificando o valor de canal\_minimo e canal\_maximo **por nome**.
- 7) Crie classes para representar estados e cidades. Cada estado tem um nome, sigla e cidades. Cada cidade tem nome e população. Escreva um programa de testes que crie três estados com algumas cidades em cada um. Exiba a população de cada estado como a soma da população de suas cidades.
- 8) Escreva uma classe Coordenada com atributos x e y, e métodos para mostrar as coordenadas, calcular a distancia para outra coordenada, comparar coordenadas, mostrar no formato coordenada polar.
- 9) Escreva classes para as seguintes formas: quadrado, retângulo e círculo.
- 10) Escreva uma classe Fracao que armazena dois inteiros, numerador e denominador.
  - a) Implemente metodos para somas, subtração, multiplicação e divisão de duas frações
  - b) Implemente o método que imprime uma fração no formato numerador / denominador
  - c) Implemente um método que inverte a fração
  - d) Implemente um método que retorna a fração em valor real
  - e) Implemente um método que cria uma fração (numerador/denominador) a partir de um número real

## Exercícios

Para cada exercício apresentado, sempre que existirem questões de interpretação ampla, genéricas, tome a liberdade de definir o escopo da aplicação em questão e como ela será utilizada. Isso o ajudará na modelagem do estado e do comportamento dos objetos (tais questões normalmente são esclarecidas com o cliente). Escreva como comentário no início de cada código, quais foram as decisões tomadas. Exemplo: *“Em meu exercício 3 a palavra genérico se refere ao baralho tradicional de 52 cartas, não englobando baralhos específicos como uno”*. Ou ainda, *“o objetivo de minha implementação é incluir qualquer baralho, seja ele de 52 cartas, baralho de uno ou baralhos de jogos específicos (card games como Magic, Pokemon e etc.)”*. Entregar as implementações para cada exercício separadamente.

### Exercício 1: Banco

O banco Tatu, moderno e eficiente, precisa de um novo programa para controlar o saldo de seus correntistas. Cada conta corrente pode ter um ou mais clientes como titular. O banco controla apenas o nome e o telefone de cada cliente. A conta corrente apresenta um saldo e uma lista de operações de saques e depósitos. Quando o cliente fizer um saque, diminuiremos o saldo da conta corrente. Quando ele fizer um depósito, aumentaremos o saldo. O banco oferece também contas especiais, com limite especial além do saldo, e conta poupança, que oferece um rendimento mensal sempre que o saldo na conta completa um mês. Evidentemente é necessário oferecer aos clientes a possibilidade de verificar saldos, extratos e um resumo com todas as informações da conta e seus respectivos clientes.

### Exercício 2 Biblioteca

Crie as classes necessárias para um sistema de gerenciamento de uma biblioteca. Os bibliotecários deverão preencher o sistema com o título do livro, os autores, o ano, a editora, a edição e o volume. A biblioteca também terá um sistema de pesquisa (outro software), portanto será necessário conseguir acessar os atributos típicos de pesquisa (nome, autor, ...).

### Exercício 3: Polinômio

Crie uma classe que modela um polinômio do tipo  $c_0 + c_1x + c_2x^2 + \dots + c_Nx^N$ . Os usuários serão estudantes de engenharia e ciências da computação que querem resolver exercícios de álgebra. A entrada são os N coeficientes em ordem crescente de potência. Os usuários precisarão acessar o grau do polinômio, avaliar o resultado para dado valor de x, somar e multiplicar dois polinômios.

--- **Desafio:** Implemente um método que plota o polinômio para um dado intervalo de entrada.

### Exercício 4: Séries

Implemente uma classe que, implementa algumas séries matemáticas importantes: Fibonacci, Fatorial, Fibonarial, Primo. Use recursão para Fibonacci e Fatorial.

### Exercício 5: Análise Combinatória

Utilizando classes para Fração e Fatorial, implemente uma classe para a realização de análise combinatória: permutação, arranjo e combinação. Implemente otimizações que permitam reduzir a carga computacional para realizar tais operações. A classe que realizará as operações de análise combinatória deve receber uma lista de elementos de tamanho (n) e um número de posições de tamanho (p). Devem ser disponibilizados métodos que retornam o número de permutas/arranjos/combinações possíveis, bem como umas das possíveis permutas/arranjos/combinações de forma aleatória.

### Exercício 6: Baralho de cartas

Projete e implemente um baralho de cartas genérico, isto é, que poderia ser usado para implementar diversos jogos de carta.

### Exercício 7: Sistema de Leitura Online

Implemente um sistema de leitura online que possua as seguintes funcionalidades:

- Criação e gerenciamento de usuários
- Busca pelos livros disponíveis (reutilizar classes do exercício 2)
- Leitura de livro (página por página)
  - Apenas um usuário por vez
  - Apenas um livro ativo por usuário

A implementação dos métodos referentes a visualização na tela (display) podem ser substituídos por um comentário dentro dos métodos, ex. `""" atualiza elementoX na tela """`

### Exercício 8: Reimplemente o jogo da forca completamente orientado a objetos

### Exercício 9: Campo minado

Projete os atributos e os métodos (sem implementação) necessários para a criação de um campo minado.

--- **Desafio:** Realize a implementação do jogo em modo textual

### Exercício 10: Organização das Turmas

Crie um sistema que gerencia o cadastro de alunos e professores em turmas. Os usuários serão os membros da secretaria. Eles devem conseguir visualizar os alunos matriculados em cada turma, com seus dados, suas notas e presenças. Além disso, os secretários precisam ter acesso a dados cadastrais dos professores associados à disciplina.

\*pesquise a respeito, e use *magic methods* do python sempre que achar adequado.