|  |
| --- |
| UNIPÊ |
|  |

Proposta de Exercícios – Disciplina: Paradigmas de Linguagens de Programação.

Lista de 15 questões sobre Programação Orientada a Objetos (POO) que podem ser implementadas em **Python**, **Java** e **Golang**:

**1. Classes e Objetos Básicos** Crie uma classe Carro com atributos como marca, modelo, e ano. Instancie três objetos dessa classe e exiba os detalhes de cada um.

**2. Métodos** Adicione um método acelerar e frear à classe Carro que altere um atributo velocidade. Crie um método para exibir a velocidade atual.

**3. Encapsulamento** Implemente uma classe ContaBancaria com atributos saldo, titular e métodos depositar e sacar. Use encapsulamento para proteger o atributo saldo.

**4. Herança** Crie uma classe base Animal com métodos como som. Derive classes como Cachorro e Gato que implementam o método som de maneiras diferentes.

**5. Polimorfismo** Utilize polimorfismo para criar uma função que receba uma lista de objetos Animal e chame o método som de cada um, demonstrando comportamento diferente para cada subclasse.

**6. Composição** Implemente uma classe Motor e associe-a a uma classe Carro. A classe Carro deve ter um objeto do tipo Motor como um de seus atributos.

**7. Associação** Crie uma classe Escola e uma classe Professor. A associação deve permitir que uma escola tenha vários professores, e um professor possa lecionar em várias escolas.

**8. Agregação** Modele uma classe Empresa que agregue uma lista de objetos Empregado. Cada empregado deve ter atributos como nome, cargo, e salario.

**9. Interfaces/Protocolos** Em Java e Golang, defina uma interface para imprimível que tenha um método imprimir. Implemente essa interface em classes como Relatório e Contrato. Em Python, use a abordagem de protocolo ou classes abstratas.

**10. Sobrecarga de Métodos (Java) / Métodos com Nomes Diferentes (Python, Go)** Implemente uma classe Calculadora com métodos somar que aceita diferentes números de parâmetros (dois ou três números). Em Golang, use funções com nomes diferentes para diferentes quantidades de parâmetros.

**11. Classes Abstratas** Crie uma classe abstrata Funcionario com um método abstrato calcularSalario. Derive classes como FuncionarioHorista e FuncionarioAssalariado que implementam calcularSalario de formas distintas.

**12. Sobrecarga de Operadores (Python) / Métodos de Conveniência (Java, Go)** Em Python, sobrecarregue o operador + para somar dois objetos Produto baseados no preço. Em Java e Golang, crie métodos que permitam essa funcionalidade.

**13. Métodos Estáticos** Adicione um método estático à classe Matematica que calcula o fatorial de um número. Em Python, utilize @staticmethod, em Java static, e em Golang crie uma função regular na struct.

**14. Singleton** Implemente o padrão de projeto Singleton para garantir que apenas uma instância de uma classe Configuracao seja criada.

**15. Exceções/Erros Personalizados** Crie uma classe de exceção personalizada SaldoInsuficienteException em Java e Python, ou error em Golang, que seja lançada quando houver uma tentativa de saque superior ao saldo disponível na classe ContaBancaria.

Essas questões abordam conceitos essenciais de POO e permitem que os alunos pratiquem a implementação em três linguagens diferentes, compreendendo as nuances de cada uma. Implemente todos os código nas linguagens Python, Java e Go..

Grave os arquivos no Github e posteriormente eu pegarei essas informações de forma individualizada de cada aluno.

Obrigado.

Atenciosamente,,  
Prof. Ricardo Roberto de Lima..