1. Explique os conceitos de Abstração, Encapsulamento, Herança e Polimorfismo em suas próprias palavras e dê exemplos de como cada conceito pode ser aplicado em um sistema de gerenciamento de biblioteca (por exemplo, classes como Livro, Usuário, Bibliotecário, etc.).

Abstração: Esconder os dados complexos e fazer com que o código fica mais simples e fácil de entender.

Ex: class Carro extends Veiculo {

@override void acelerar() {

print("O carro está acelerando."); }

}

void main() {

Veiculo meuCarro = Carro();

meuCarro.acelerar();

}

Encapsulamento: Esconder os dados internos de uma classe.

EX: class Pessoa {

String \_nome;

Pessoa(this.\_nome);

String obterNome() {

return \_nome;

}

void definirNome(String nome) {

\_nome = nome;

}

}

void main() {

var pessoa = Pessoa('João');

print(pessoa.obterNome());

pessoa.definirNome('Maria');

print(pessoa.obterNome());

}

Herança: A classe filha herda classes e métodos da classe pai.

EX: class Animal {

void fazerSom() {

print("O animal faz um som.");

}

}

class Cachorro extends Animal {

@override

void fazerSom() {

print("O cachorro faz: Au au!");

}

void abanarRabo() {

print("O cachorro está abanando o rabo.");

}

}

void main() {

var meuCachorro = Cachorro();

meuCachorro.fazerSom(); // Saída: O cachorro faz: Au au!

meuCachorro.abanarRabo(); // Saída: O cachorro está abanando o rabo.

}

Polimorfismo: Ele reutiliza o nome em classes diferentes.

EX: abstract class Forma {

void desenhar(); // Método abstrato que deve ser sobrescrito

}

class Circulo extends Forma {

@override

void desenhar() {

print("Desenhando um círculo.");

}

}

class Quadrado extends Forma {

@override

void desenhar() {

print("Desenhando um quadrado.");

}

}

void main() {

List<Forma> formas = [Circulo(), Quadrado()];

for (var forma in formas) {

forma.desenhar();

}

}

1. Pergunta: Quais são as principais diferenças entre a Programação Orientada a Objetos (POO) e a Programação Funcional? Cite vantagens e desvantagens de cada abordagem.

O POO carrega os conceitos da herança, poliforfismo, encapsulamento e abstração. Vantagens: Modularidade, reusabilidade, manutenibilidade e modelagem natural. Desvantagem: Complexidade, desempenho, dependências.

Programação funcional tem os conceitos de funções como cidadãos de primeira classe, imputabilidade, funções puras. Vantagens: Previsibilidade, concorrência, reusabilidade, testabilidade. Desvantagem: Curva de aprendizado, verbosidade e desempenho.