

Atividades de POO

1. Sistema de Controle de Estoque

Crie uma classe `Produto` com atributos como `id`, `nome`, `quantidade` e `preco`. Implemente métodos para adicionar e remover quantidades do estoque, ajustando de acordo com as operações realizadas.

2. PDV (Ponto de Venda)

Desenvolva uma classe `ItemVenda` que contém `produto`, `quantidade` e `precoUnitario`. A classe `Venda` deve agregar múltiplos `ItemVenda` e calcular o total da venda.

3. Sistema de Agendamento de Consultas

Implemente uma classe `Paciente` com informações básicas e uma classe `Consulta` que herda as informações do `Paciente` e adiciona detalhes da consulta, como `data`, `hora` e `especialidade`.

4. Biblioteca de Músicas

Crie uma classe `Musica` com `titulo`, `artista` e `duracao`. A classe `Playlist` deve conter uma lista de `Musica` e métodos para adicionar e remover músicas, além de calcular a duração total da playlist.

5. Gerenciamento de Funcionários

Desenvolva uma classe `Funcionario` com atributos como `id`, `nome`, `cargo` e `salario`. Crie métodos para promover o funcionário (aumento de salário) e para alterar o cargo.

6. Sistema de Reservas de Hotéis

Implemente uma classe `Quarto` com `numero`, `tipo`, `preco` e `reservado` (booleano). A classe `Reserva` deve manipular reservas, alterando o estado de `reservado` dos quartos.

7. Cadastro de Veículos

Crie uma classe `Veiculo` com atributos como `placa`, `modelo`, `ano` e `cor`. Implemente métodos para registro e atualização dos dados do veículo.

8. Sistema de Matrículas Escolares

Desenvolva uma classe `Estudante` com `nome`, `matricula` e `curso`. A classe `Matricula` deve associar `Estudante` a classes específicas, gerenciando as matrículas.

9. Controle de Acesso

Implemente uma classe `Usuario` com `nome`, `email` e `senha`. Crie uma classe `Sistema` que gerencia usuários, incluindo registro e autenticação baseada em email e senha.

10. Gerenciamento de Projetos

Crie uma classe `Projeto` com `nome`, `descricao`, `dataInicio` e `dataFim`. A classe `Tarefa` associa-se a um `Projeto` e contém `descricao`, `responsavel` e `prazo`.

11. Sistema de Pedidos de Restaurante

Desenvolva uma classe `ItemMenu` com `nome` e `preco`. A classe `Pedido` deve conter uma lista de `ItemMenu`, calcular o total do pedido e registrar a mesa que realizou o pedido.

12. Sistema de Avaliações de Produtos

Implemente uma classe `Avaliacao` com `produto`, `usuario`, `nota` e `comentario`. A classe `Produto` deve ser capaz de calcular sua média de avaliações.

13. Sistema de Controle de Tarefas

Crie uma classe `Tarefa` com `descricao`, `prioridade` e `concluida` (booleano). Implemente métodos para marcar a tarefa como concluída e para alterar sua prioridade.

14. Sistema de Gerenciamento de Cursos

Desenvolva uma classe `Curso` com `nome`, `descricao`, `cargaHoraria` e uma lista de `Estudantes`. Implemente métodos para adicionar e remover estudantes do curso.

15. Controle de Vendas Online

Implemente uma classe `Produto` com `nome`, `descricao`, `preco` e `estoque`. A classe `VendaOnline` deve registrar vendas, atualizar estoque e calcular o total vendido.

16. Sistema de Agendamento de Serviços

Crie uma classe `Servico` com `descricao`, `preco` e `duracao`. A classe `Agendamento` deve registrar agendamentos, associando um `Servico` a um horário específico.

17. Sistema de Gerenciamento de Biblioteca

Desenvolva uma classe `Livro` com `titulo`, `autor`, `ano` e `disponivel` (booleano). A classe `Emprestimo` gerencia o empréstimo de livros, alterando o status de disponibilidade.

18. Sistema de Gerenciamento de Contas a Pagar

Implemente uma classe `Conta` com `descricao`, `valor`, `vencimento` e `paga` (booleano). Crie métodos para registrar o pagamento das contas, alterando seu status.

19. Sistema de Cadastro de Eventos

Crie uma classe `Evento` com `nome`, `local`, `data` e `participantes` (lista de nomes). Implemente métodos para adicionar e remover participantes do evento.

20. Sistema de Monitoramento de Sensores

Desenvolva uma classe `Sensor` com `tipo`, `localizacao` e `valorAtual`. A classe `Monitoramento` deve registrar leituras dos sensores e alertar para valores fora do esperado.

Cada atividade é projetada para ser um mini-projeto onde você pode aplicar conceitos de POO, praticando Herança, Encapsulamento, e o uso de Classes em cenários reais de desenvolvimento de software com JavaScript e Node.js.