

1. Faça um resumo/tabela sobre alguns paradigmas de programação e cite as principais características de cada um deles. Cite também linguagens de cada um dos paradigmas

Paradigma Imperativo

Instruções devem ser passadas ao computador na sequência que devem ser executadas, com o foco da resolução do problema usando linguagens procedurais sendo a maneira como deve ser feita. Útil para projetos em que não se espera que haja mudanças significativas, programas mais estáticos, para operações complexas ou quando houverem muitos elementos compartilhados. Códigos eficientes e flexíveis, mas, em geral, menos legível. Linguagens: Cobol, Fortran, Pascal.

Programação Procedural

Muito usada para programação de uso geral, é uma lista de instruções para informar o computador o que fazer, podendo ser agrupadas em procedimentos que podem ser reutilizados. Linguagens: C, C++, Java, Pascal.

Programação Orientada a Objetos

Torna o código mais modular por poder associar problemas reais diretamente com o código, construindo objetos com comportamentos, atributos e estados, podendo refletir diretamente aquilo que se objetiva construir. Útil para construir grandes sistemas, muito código a ser usado novamente e compartilhado e previsão de mudanças no projeto. Linguagens: C++, Java, Ruby, Python, C#.

Programação Paralela

Permite uso de múltiplos processadores para execução de código, dividindo o trabalho entre eles para chegar em um mesmo objetivo. Linguagens: C, C++, Elixir.

Paradigma Declarativo

Foca naquilo que será resolvido e não necessariamente em como, contando com maior nível de abstração, há declarações iniciais de verdades lógicas imutáveis e, depois de realizar interações entre elas, o resultado encontrado sempre será igual para as declarações feitas. Há facilidade no acesso ao banco de dados e maior abstração da solução implementada em código, resultando em códigos menores.

Paradigma de Lógica de Programação

Baseado em fatos e usado que sabe para criar cenário onde todos são verdadeiros e apontam para algum resultado final, podendo chegar a conclusões lógicas baseadas em tais apontamentos. Muito usado para aplicações de Inteligência Artificial mas também para comprovar teoremas e criar programas especialistas. Linguagens: Abys, Ciao, Alice, Prolog.

Paradigma Funcional

Baseia-se no uso de funções matemáticas, logo, pequenas funções formam o programa, com código definido dentro delas e variáveis no escopo definido para a função. Funções não alteram valores fora do escopo dela e funções não são afetadas por valores fora do escopo. Linguagens: Haskell, Scala, Racket.

2. *Pesquise sobre outras linguagens de programação existentes e seus respectivos paradigmas.*

Rust: linguagem multi-paradigma, incluindo concorrente, funcional, imperativo e estruturado. Enfatiza performance, segurança de tipo, concorrência e segurança de memória, com todas referências apontando para endereços válidos, sem o uso de coletor de lixo, usado de um mecanismo de “borrow checker” para verificar o tempo de vida das referências na compilação do programa. Usado por empresas como Amazon, Discord, Meta e, recentemente, no kernel do sistema Linux.

Lua: linguagem de programação feita por brasileiros, é também multi-paradigmas, podendo ser usada para scripts, programação imperativa, baseada em protótipos e orientada a objetos e funcional. Focada em velocidade, portabilidade e extensão.

Erlang: linguagem concorrente e funcional de alto nível, com coletor de lixo em tempo de execução, feita para sistemas distribuídos, tolerantes à falha, parcialmente de tempo real, alta disponibilidade e quando o código pode ser alterado sem parar completamente o sistema, muito usada para redes de telecomunicação e aplicações como Whatsapp.

3. *O que é o paradigma de programação orientado a agentes? Pesquise sobre eles e cite algumas linguagens*

Arquitetura de software na qual entidades computacionais de maior interesse (“agentes”), iniciado com IA distribuída, com principal característica sendo que os agentes são entidades autônomas com objetivo próprio, dado pelo projetista, mas com o agente visando o atingir da melhor forma que conseguir. Usado para controle e automação, como sistemas de controle de transporte, aceleradores de partícula e controle de tráfego aéreo, telecomunicações, jogos e interatividade

Há frameworks que implementam tal paradigma, como para Java e C++. As linguagens AgentSpeak e SARL são implementadas no paradigma orientado a agentes.