Liguagens, Paradigmas de programação e máquinas virtuais

Linguagem de Programação – são utilizadas para traduzir o que queremos que nosso aplicativo faça, temos que ser muito específicos, pois os sistemas não tem inteligência para interpretar coisas ambíguas. Portanto, as instruções que damos, compostas por uma serie de etapas chamadas algoritmos, são muito importantes.

Essas instruções são dadas por meio de uma linguagem formal, que não permite que haja varias interpretações sobre o que vai ser transmitido, para que possamos ser sempre claros e precisos. As linguagens formais que usamos para fornecer essas instruções são chamadas de “linguagens de programação”.

Geralmente encontramos dois grandes grupos:

- Específicos: Resolvem problemas específicos. Como exemplo, podemos citar uma linguagem de programação que faça gráficos matemáticos.

- Gerais: Permitem desenvolver um numero infinito de diferentes aplicações de qualquer contexto, como um site de animais de estimação ou um e-commerce.

Em programação, temos linguagens de alto e baixo nível, as de alto nível são aquelas q estão mais próximas da linguagem natural do que a linguagem de maquina (zero e uns). Elas nos permitem escrever código de uma forma mais natural e rápida para focar em funcionalidades interessantes. As de baixo nível, embora seja mais fácil programarmos um uma linguagem de alto nível, as linguagem de baixo nível são usadas para fornecer instruçies muito especificas, e para aproveitar ao máximo os recursos disponíveis.

Tipos de linguagem e frameworks

Linguagens fortemente tipadas e fracamente tipadas são diferenciadas de acordo com o fato de permitirem ou não violações de tipos de dados em variáveis declaradas

Tipagem fraca – nessas linguagens não indicamos na maioria das vezes, o tipo de variável. Aqui podemos atribuir, por exemplo, um valor inteiro a uma variável que anteriormente tinha uma string. Mas, não só isso. Tambem podemos operar com variáveis de diferentes tipos.

Sua principal vantagem é que é muito mais rápida de se desenvolver, mas uma clara desvantagem é que podemos cometer muito mais erros se não tivermos cuidado.

Tipagem forte – Nessas loinguagems, somos obrigados a indicar o tipo de dado ao declarar a variável. Alem disso, este tipo não pode ser alterado uma vez que a variável é definida. Sua vantagem é que, por ser um código mais expressivo, cometemos menos erros. A desvantagem é que eles são muito mais rígidos durante a programação, e vc precisa escrever muito mais código.

Tipagem estática – NA tipagem estática, a verificação de tipos é feita durante a compilação e não durante a execução. Em comparação com a tipagem dinâmica, a tipagem estática permite que os erros de digitação sejam detectados mais cedo, e a execução do programa seja mais eficiente e segura.

Tipagem dinâmica – A verificação de tipos é feita durasnte a execução e não durante a compilação. Comaprado a tipagem estática é mais flexível, apesar de funcionar mias lentamente e ser mais propensa a bugs.

Diagrama

Descrição gerada automaticamente com confiança baixa

FRAMEWORKS

É uma estrutura previa/esqueleto, que pode ser usada para desenvolver um projeto. O Framework é uma esoecie de tamplate, um esquema conceitual, que simplifica a preparação de uma tarefa, pois basta complemente-ala de acordo com o que se deseja fazer.

Paradigmas de Programação

Decada de 1960 – A forma de programação era estruturada, e as linguagens de programação seguiam esse modelo ou paradigma, em que era preciso escrever funções e o código era executado linha após linha

As simulações eram extremamente complexas, e abordá-las com o modelo de programação estruturada adicionava complexidade ao invés de facilitar o desenvolvimento

Entao foi decidido mudar o modelo, escrever o código, mas com outro conjunto de regras, em que pudesse representar literalmente os diferentes objetos que compõem um navio. Cada objetp tinha um estado e funções internas, assom como uma nave pode ter um estado ligado e desligado, e uma função que permite que seja ligado ou desligado.

A linguagem de programação SIMULA 67 e o paradigma de programação orientada a objetos, nasceram lá.

Existem vários paradigmas e alguns não dependem da linguagem, mas são uma filosofia de como escrever código e quais regras seguir nesse modelo.

Porem, existem alguns paradigmas que se destacam pela sua utilização e filosofia, tais como:

- ESTRUTURADO – Segue uma linha de pesamento em que uma instrução geralmente é executada por vez, e a outra é governada por um conjunto limitado de instruções.

- ORIENTADO A OBJETOS – Na programação orientada a objetos, o código pode ser agrupado de forma qur represente uma entidade e interprete mensagens.

- FUNCIONAL – O paradigma de programação funcional é baseado em um conceito muito smples: o de funções matemáticas. A forca desse paradigma é q sempre que a função X recebe o valor A, ela sempre retorna o valor B. Essa propriedade de retornar o mesmo valor é conhecida como imutabilidade, sendo característica desse paradigma.

- LOGICO – na programação logica em vez de desenvolver etapas e instruções, ele usa regras logicas para consultar o sistema e inefere o que fazer com base nas regras logicas estabelecidas.

- PARADIGMA DE PROGRAMACAO COM LINGUAGEM ESPECIFICA DE DOMINIO – No paradigma de programação de domínio especifico, as linguagens que encontramos aqui tentam resolver problemas muito específicos. Ex. qndo queremos consultar a base de daos de um supermercado para saber quais os produtos que temos nas categoria de eletromésticos.

- MULTIPARADIGMA – Ao longo da evolução da programação, com novos desafios e paradigmas, houve linguagens que modificaram a sua estrutura para fornecer soluções em diferentes paradigmas. Um exemplo dessas linguagens é o javascript, onde se pode escrever código de programação estruturadp, orientado a objetos e ate mesmo estrutural

A melhor maneira de conhecer um paradigma de programação é pesquisar e programar em uma linguagem característica desse paradigma. Voce não precisa ser um especialista, mas deve estar aberto a conhecer novas ferramentas.

Paradigmas – Imperativo

Procedural

Estrutura de blocos

Orientação a objetos

Computação distribuída

- Declarativo

Funcional

Programação lógica

Do código ao executável

Código Fonte – Codigo fonte pe uma coleção de instruções de computador, escritas usando uma linguagem de programação legível

Código de maquina – é uma sequencua de instruções m linguagem de maquina ou binaria. È o resultado obtido depois que o compilador converte o código fonte em uma linguagem que pode ser entendida pelo processador

Compilador – é um aplicativo que traduz (compila) o código fonte em um código que o processador pode entender e executar. Este código de maquina é armazenado na forma de um arquivo executável.

Interceptador – ele traduz o código-fonte linha por linha e o executa diretamente. O processo de tradução funciona muito mais rápido do que em um compilador, mas a execução é mais lenta e requer uma grande qntd de memoria

Introducao a maquinas virtuais

Servidores geralmente usavam apenas 30% de sua caoacidade total, então os outros 70% eram constantemente dsperdicados. Para resolver estes problemas de uso indevido de recursos, foi criada a virtualização que nos permite estar no mesmo computador rodando vários sistemas operacionas que fingem ser computadores reais.

É um software capaz de conter um sistema operacional no interior, fazendo parecer um computador real e este outro sistema operacional pode abrigar outro...

Existem 2 tipos de sistemas virtuais

Os de sistema operacionais – emula um computador completo, o local onde a maquina virtual é instalado é chamado hipervisor, que uma camada, instalada na parte física do computador, e sua função é atribuir, parte da memoria, disco rígido e outros recursos físicos.

Existem 2 tipos de hiper visor, o primeiro é mais utilizado por ser mais rápido e seguro, é executado dieramente na parte física do computador, e sera criado nele 1 ou mais maquinas virtuais

Texto

Descrição gerada automaticamente

O segundo tipo de hipervisor é executado dentro do sistema operacional e é mais lento q o anterior

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Sobre o hipervisor, podemos criar qntas maquinas virtuais quisermos

Os de processos – n emulam um computador completo, apenas um processo especifico. Ex; um aplicativo, permitindo q cada um se comporte da mesma forma independete do sistema operacional q sera executado. Que é muito útil no desenvolvimento de aplicativos, q sera utilizado em diferentes sistemas operacionais.

Virtualização

- emulação de ambientes isolados, capazes de rodar diferentes sistemas operacionais dentro de uma mesma maquina

O q são maquinas virtuais

-

Containers e orquestradores

Cada vez mais organizações estão usando containes em seus aplicativos

- O q são container (ex: Docker)

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média

O software de orquestacao de containers permite que os desenvolvedores usem vários containers para a impementacao de seus asplicativos. Conforme as empresas executam seus aplicativos em containers e os movem para nuvem, há uma demanda crescente por soluções de container orchestation.

Para ser incluído na categoria de orquestacai de containers, um software deve:

- Permitir que os administradores provisiones hosts,

- programar e automatizar a implantação de containers

- executar varias instancias de containers

- alertar os usarios sobre cotainers com falha

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente