

Aluno(a): Brenda Gabriela Martinez Araujo

Curso: Análise e Desenvolvimento de Sistemas 2022.2

Matéria: Linguagem de Programação

**1 – Quais são as diferenças entre uma linguagem de programação compilada e uma linguagem de programação interpretada?**

Uma linguagem compilada utiliza um compilador, já presente na máquina em que foi desenvolvido o programa, para executar o código escrito. Para isso, ao executar o código, primeiro deve-se compilar manualmente o programa inteiro (build). Já as linguagens interpretadas, utilizam um interpretador externo à máquina de origem do código, precisando adicioná-lo ao computador. Esse, lê cada linha do código e o executa, traduzindo o código em tempo de execução.

**2 – O que são scripts?**

Scripts são códigos que contém uma série de instruções, geralmente sendo repetidas em um contexto específico, como automatização de tarefas laboriosas, contabilização de usuários e ações em um website, customização, etc.

**3 – Como funciona o Java (compilação e execução do código)?**

Para executar um código em Java, é necessário a utilização de um software, que funciona como um segundo computador, chamado Java Virtual Machine. A JVM é uma forma de garantir que os programas escritos em Java funcionem em qualquer SO e hardware. O compilador Java primeiro converte o código-fonte em bytecodes, que serão traduzidos posteriormente para linguagem de máquina em sua execução, através da JVM.

**4 – Como funciona o Python (compilação e execução do código)?**

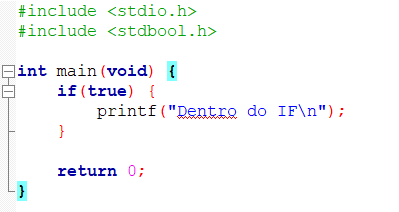
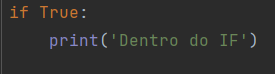
Python é considerada uma linguagem interpretada, pois apesar de traduzir o código-fonte em bytecodes (assim como em Java), ela utiliza o interpretador CPython para fazer a execução das instruções em tempo real, ou seja, a medida em que é feita a leitura de cada linha.

**5 – O que são: Análise léxica, análise semântica, análise sintática e autômatos finitos (máquinas de estado)?**

Análise léxica é a leitura do programa caractere a caractere, de modo a agrupar os caracteres em tokens. A partir desses tokens, é feita a análise sintática, na qual o analisador cria uma “árvore sintática” (estrutura de dados) a partir de regras pré-estabelecidas da linguagem de programação. Na análise semântica, verifica-se os erros semânticos no código-fonte (por exemplo expressões matemáticas que resultem em erro, tipos, variáveis, etc).  
 Máquina de estados finita é um modelo matemático que representa programas de computadores ou circuitos lógicos como uma máquina abstrata que deve estar em um dos seus finitos estados, chamado de estado atual. Um estado representa um nó de comportamento do sistema esperando uma condição ou evento para executar um conjunto de ações. Na computação, são utilizadas na engenharia de software, compiladores, protocolos de rede, etc.

**6 – Quais as diferenças de um IF em C e em Python? Como se sabe onde inicia e quando termina a lógica do IF em cada linguagem?**

A diferença de um IF em C e em Python está na sintaxe da estrutura condicional. Enquanto em C é necessário que a condição esteja entre parênteses e as instruções entre chaves, para o Python, o que vai valer é a identação das instruções, que devem estar abaixo do IF, com um espaçamento de acordo com a especificação da linguagem.

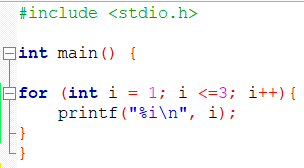
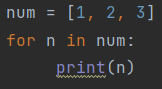


*Estrutura condicional if em C*

*Estrutura condicional if em PYTHON*

**7 – Quais as diferenças de um FOR em C e em Python?**

Em C, o FOR exige que haja um valor inicial em uma variável, uma condição para finalização do laço e um valor para incremento (ou decremento) dessa variável. Já em Python, o FOR percorre os itens de uma coleção e a variável indicada no próprio for vai receber o item atual da coleção.



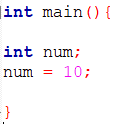
*Ambas retornam:  
1  
2  
3*

*Laço FOR em Python*

*Laço FOR em C*

**8 – Como é a declaração e o uso de variáveis em Perl, Python e C?**

Em C a declaração de uma variável é obrigatória, sendo necessário declarar o seu tipo e seu nome de acordo com as regras da linguagem. Já em Perl, não é necessário declarar as variáveis, pois essas são geradas dinamicamente conforme as utilizamos no código. Por fim, em Python, apesar de não ser obrigatório, é possível declarar uma variável atribuindo um valor a mesma, porém não é preciso atribuir um tipo, que é identificado de acordo com o valor atribuído.



*Declarando uma variável*

*em Python*

*Declarando uma variável*

*em Perl*

*Declarando uma variável*

*em C*

**9 – Existem objetivos específicos na criação das linguagens de programação? Comente sobre o Delphi, Python, R, C e C++. Indicando os criadores e alguns dos seus objetivos.**

Sim, pois cada linguagem de programação é criada para suprir uma necessidade.

A linguagem Delphi foi criada pela empresa Borland em 1995, baseada em *object pascal*, com o objetivo de fornecer uma plataforma de programação para desktop simples, mas com recursos avançados para o desenvolvimento no Windows. Hoje, ainda é utilizada em websites, machine learning, e até para o desenvolvimento de aplicativos móveis.

Já o Python, foi criada por hobby pelo programador Guido Van Rossum, com o objetivo de ser uma linguagem fácil para iniciantes aprenderem, baseada na linguagem BASIC. Hoje, o Python é uma das linguagens mais utilizadas do mundo, sendo útil em quase todas as áreas, incluindo websites, scripts, inteligência artificial, machine learning, etc.

A linguagem R foi criada pelos professores de estatística Ross Ihaka e Robert Gentleman, que estavam insatisfeitos com as opções de software para análise de dados disponíveis na época. Então começaram a desenvolver a linguagem R a partir da linguagem S, uma linguagem de programação comercial para análise de dados. Atualmente a linguagem R tornou-se independente e muito utilizada para análise de dados e machine learning.

A linguagem C foi criada em 1972 por Dennis Ritchie, nos laboratórios Bell da AT&T. Dennis precisou criar uma linguagem eficiente, portável e flexível para desenvolver sistemas operacionais numa época em que os SOs exigiam amplo conhecimento de hardware. A linguagem C surgiu como um aperfeiçoamento da linguagem ALGOL e B, de Ken Thompson.

C++ foi criada por Bjarne Stroustrup em 1983 nos laboratórios Bell como um aperfeiçoamento da linguagem C, adicionando recursos de orientação a objetos. Seu desenvolvimento foi necessário pois os sistemas complexos que utilizavam a linguagem C eram difíceis de controlar devido à alta quantidade de linhas de código (de 25.000 a 100.000). Com as características da orientação a objetos na linguagem, o controle, o uso, e o entendimento do código tornou a programação de programas complexos mais eficaz.

**10 – O que são linguagens de alto nível e de baixo nível? Cite exemplos e o porquê.**

Linguagens de baixo nível são as chamadas “linguagens de máquina”, onde é necessário um alto conhecimento de hardware e registradores, como a Assembly. Já as linguagens de alto nível possuem um nível maior de abstração, que as aproximam mais da língua dos humanos. São as mais usadas atualmente pois permite que o programador não precise se preocupar com conceitos complexos de hardware, como por exemplo Python e C.

**11 – Quais as diferenças entre as classificações de linguagens: tipagem forte, tipagem fraca, estaticamente tipada, dinamicamente tipada...? Cite exemplos e o porquê (Exemplos de linguagens classificadas como fortes ou fracas... e o porquê elas são classificadas assim).**

Tipagem forte acontece quando a linguagem impõe regras rigorosas sobre o uso dos tipos de dados, sendo necessária a conversão do tipo do dado (int, char, string) para que seja passado como outro tipo. Por exemplo, na linguagem Java, se quiser utilizar uma string ‘10’ como um tipo inteiro, é preciso utilizar uma função que converta o dado em INT.

Já na tipagem fraca, essa restrição não é obrigatória, então é possível, por exemplo, passar um número inteiro numa string sem ser necessário convertê-lo. Algumas linguagens de script, como Javascript e Perl, geralmente são consideradas linguagens de tipagem fraca.

Uma linguagem é considerada estatisticamente tipada se verifica erros na tipagem no momento da compilação, como por exemplo na linguagem C. Caso haja um erro na tipagem dos dados, o programador é alertado sobre o erro e pode consertá-lo, evitando bugs.

Uma linguagem dinamicamente tipada verifica os erros da tipagem no momento da execução do código, isso permite uma utilização mais flexível do código, porém só é possível detectar erros na tipagem dos dados no momento da execução do código, e o programador deve ter uma atenção maior no tipo dos dados a fim de evitar bugs. Um exemplo de linguagem dinamicamente tipada é a linguagem interpretada Python.

**12 – O que são linguagens orientadas a objetos? O que são objetos? O que são classes, hierarquia, polimorfismo, herança, encapsulamento, abstração e os modificadores de acesso (public, private e protected)?**

As linguagens orientadas a objetos utilizam objetos, que representam entidades do mundo real, que interagem entre si para tornar a programação de sistemas complexos mais eficiente.

Por exemplo, o tipo de um objeto em POO é definido por classes, que contém atributos e métodos para definir seus dados e comportamentos.

Hierarquia define a relação de herança entre classes, classificando a herdeira como subclasse e a classe base como superclasse.

O polimorfismo permite que um objeto possa ser utilizado de diferentes formas em uma hierarquia de classes, podendo implementar métodos em outra classe sem necessariamente fazer implementações específicas para cada classe. Essa relação diminui a quantidade de código escrito e torna a depuração do código mais simples.

A herança permite que uma classe herde propriedades e métodos de uma outra classe.

O encapsulamento esconde atributos da classe para que essa não seja usada de forma errada, através dos métodos get e set.

Abstração é a capacidade de representar apenas os detalhes importantes do objeto, por exemplo, é possível criar uma classe abstrata para definir métodos e propriedades comuns a certos objetos.

Por fim, os modificadores de acesso definem onde podem ser acessadas as propriedades e métodos de uma classe, por exemplo, o modificador public permite que sejam acessados de qualquer lugar, o private apenas dentro da própria classe e o protected dentro da própria classe e suas subclasses.