Anotações sobre Bootcamp do Santander

- Pensamento computacional -

Pensamento computacional? É um pensamento que envolve passos computacionais ou algoritmos que podem ser implementados no computador.

Esse pensamento é uma habilidade generalista que pode ser aplicada a diversas áreas do conhecimento.

- é uma forma de pensamento logico para solução de problemas que podem ser solucionados pelo ser humano ou pelo computador, ou até mesmo pelos dois em conjunto –

O pensamento computacional está divido em 4 pilares: decomposição, reconhecimento de padrões, abstração e designer de algoritmos.

* Decomposição = dividir as situação em exemplos menores para resolução e analise,
* Reconhecimento de padrões = identificar padrões e tendências ou similaridades e diferenças entre situação. Ex: algo que se repete sempre que determinada ação é tomada, algo que reproduz um tamanho especifico de acordo com a posição,
* Abstração = Pensar de forma generalista algum tipo de resolução que se adapte para qualquer problema,
* Designer de algoritmos = Definição de passo a passo para a solução de problemas, automatização.

Um processo continuo deve conter as etapas de definição de problema, testar soluções e aperfeiçoar a solução adequada.

Técnicas da lógica de programação:

* Técnica linear: é uma sequência de passos bem definidos para resolução do problema;
* Técnica estruturada: possui o mesmo princípio da linear porem pode contém algumas opção para chegar até o resultado que se busca;
* Técnica modular: é uma técnica que decompõem o problema em módulos (várias partes e regras a serem seguidas) e busca a resolução de cada problema separado para chegar ao resultado completo de forma geral;

- FUNDAMENTOS DE ALGORITMOS –

Tipos de dados existentes = numérico, caracteres e lógico.

* Numéricos: Inteiros (todos os números positivos e negativos que não possuem casas decimais) ex: 1.. 20.. 30.. -50.. -800.. 15 e etc.

Reais (todos os números positivos e negativos porem

possuem casas decimais) ex: 1,15.. 25,31.. 0,1.. 3,26.. -50,13 e etc.

* Caracteres: possui um tamanho pré definido antes da sua entrada e pode ser representado por qualquer letra, símbolo ou número.
* Lógico: é representado por verdadeiro (0) ou falso (1), que está relacionado a lógica BOOLEANO onde poderá ser representado por .verdadeiro, .v ou .s.... e .falso, .f ou .n

Uma variável pode assumir qualquer um dos valores de um determinado conjunto de valores. Sendo assim ela poderá ter dois papeis definidos, onde ela poderá ser de Ação (onde modifica o estado de algo) ou de Controle (onde ela controla, vigia ou armazena algum determinado dado)

A principal diferença para uma constante é que na constante ela recebe um dado que permanece inalterável.

Instruções primitivas – são as ações determinadas que vamos executar em cima dos dados, geralmente cálculos matemáticos.

Estrutura condicional – quando existi um parâmetro de condição para algo ser validado, ex: se for > 7 ok se for < 7 erro.

Estruturas de repetição – são estruturas de looping, ex: for, forEch, while. Nessa condição o código se torna reduzido e mais dinâmica em questão de manutenção.

Vetores e matrizes – vetor (é uma variável dimensionada com um tamanho pré fixado), matrix (é uma tabela organizada em linhas e colunas M = linhas X N = colunas). Obs: arrays trata-se de um tipo de vetor

Instruções de entrada / saída = entrada (são dados imputados para alguma determinada ação) saída (pode ser a resolução do dado imputado ou a impressão de algum determinado erro ou bug)

- LINGUAGEM DE PROGRAMAÇÃO –

O computador entende um programa através de um código fonte que pode passar por dois processo durante esse entendimento, tradução e interpretação.

No processo de tradução o computador encaminha o programa para um compilador que vai analisar o programa e traduzir em uma linguagem de máquina para que o entendimento seja mais fácil. Nesse caso o programa de linguagem de alto nível depois que passa pelo compilador se transforma em uma linguagem objetivo e encaminhada para o entendimento do computador.

No processo de interpretação o programa fonte é executado diretamente, porem o processo se torna um pouco mais lenta.

Com a tradução temos então uma execução mais rápida e programas menores. Na interpretação gera maior flexibilidade senão mais fácil a programação, porem o processo é mais lento.

Existem algumas característica importantes na hora da criação de um programa como: legibilidade, redigibilidade, confiabilidade e custo.

* Legibilidade: facilidade de leitura, compreensão, coerência nas instruções e definição adequada das estruturas.
* Redigibilidade: facilidade da escrita do código, coerência nas instruções, suporte à abstração, reuso do código, expressividade.
* Confiabilidade: verificação de tipos, trata exceções, uso de ponteiros, compatibilidade entre compiladores (fazer o que foi programado para fazer).
* Custo: treinamento, codificação, compilação, execução e infra-estrutura.

Outras características que devemos avaliar são: atualizações, uso para IA, disponibilidade de ferramentas, comunidade ativa, adoção pelo mercado.

As análises que o compilador realiza do programa são:

* Analise léxica: particiona, clássica e elimina. Nesse processo ele analisa letra por letra classificando e transformando em tokens e elimina espaços em branco e comentários.
* Analise sintática: analisa a forma da linguística utilizada durante a programação, cada linguagem possui um padrão sintático especifico.
* Analise semântica: analisa o significado dos operadores utilizados de acordo com a regra de cada linguem.

- PORTUGOL -

Verificar no Google Portugol Web Studio – possui alguns exemplos de logica com esse formato e possui editor para realizar testes.