

Nome: Matheus Leoni de Silva Barbosa
Matrícula: 2023333TADSO346

3. Decomposição: Quebrar o problema em partes (etapas / subproblemas)

Identificação de Padrões: Observar tendências, repetições, causas e consequências que se repetem.

Abstração: Agrupar situações por meio de (~~representações~~) representações de grupo genéricas aplicáveis em problemas semelhantes.

Algoritmos: Passo a passo aplicável ao contexto estudado que "resolve" o problema (todas as instâncias).

Programação: É aplicar o conjunto de ordens em contexto de linguagens de programação / Enunciado.

2)

a) Expressão é uma unidade de código que pode ser avaliada (Ex: $2 + 2$)

Instrução é a unidade de código ~~com~~ ~~feito~~ que é feito dada a ordem para o computador

b) A expressão de lógica booleana é uma estrutura que pode ser verdadeiro ou falso. As expressões podem ser ligadas através de operadores (AND, OR, NOR) para possibilitar a criação de expressões complexas.

c) Variável: É um objeto capaz de reter e representar um valor ou expressão.

Valor: São os dados atribuídos a variável (como número ou letras)

Tipos: é uma combinação de valores e de operações que uma variável pode executar

d) O computador entende apenas 0 e 1, então por isso criamos a linguagem de alto nível e você consegue visualizar o código usando compiladores e interpretadores para transformar as instruções em 0 e 1.

e) Fluxo de dados é o (caminho) caminho que as informações percorrem em um computador. ^{isto}Entrada, processamento e saída

Entrada: Serve para os dispositivos que podemos conectar ao computador para realizar a transmissão de dados

Processamento: serve para organizar de forma adequada os dados de entrada, e para serem convertidos à saída

Saída: é caracterizado pelos equipamentos que mostram os dados

3) a) Funções é um conjunto nomeado de instruções sequenciadas, fragmentando a solução, modularizando o código, aumentando a qualidade do código.

b) Estruturas agrupam o conjunto de comandos, que são executados quando o procedimento é chamado. Elementos seriam as expressões condicionais e os intrínsecos.

c) Facilidade de leitura, pois um grupo de instruções passa a ser nomeado, ter significado explícito

Redução de linhas de códigos, devido ao reaproveitamento em vez de repetição

Facilidade de depuração, ~~(pela)~~ pelo isolamento das funcionalidades

Reutilização no mesmo arquivo ou em outros projetos

9) Operadores Relacionais ($==$, $!=$, $>$, $<$, $>=$, $<=$)

Os operadores relacionais são usados em comparações, os operandos podem ser numéricos, strings, lógicos ou objetos

$==$, Estritamente igual, retorna verdadeiro caso os operandos sejam iguais e do mesmo tipo

$!=$, Não igual, Retorna verdadeiro caso os operandos não sejam iguais e/ou não sejam do mesmo tipo

$>$, maior que, verdadeiro caso o operando do esquerdo seja maior

$<$, menor que, verdadeiro caso o operando do esquerdo seja menor

$>=$, maior que ou igual, verdadeiro caso seja maior ou igual ao do direito

$<=$, menor que, ~~igual~~ ou igual, verdadeiro, caso seja o operando do esquerdo seja menor ou ~~igual~~ igual ao do direito

Operadores lógicos ($\&\&$, $--$, $||$, $!$) Permitem compor variáveis retornando valores booleanos (Falso ou verdadeiro.)

$\&\&$, AND ($\&$), retorna ~~se~~ verdadeiro caso ambos operandos sejam verdadeiros, ~~caso~~ contrário, retorna falso.

$||$, OU, retorna verdadeiro caso ambos os operandos sejam verdadeiros; se ambos forem falsos, retorna falso.

$!$, NOT, Retorna falso caso o único operando possa ser convertida para verdadeiro, senão, retorna verdadeiro.

5. As estruturas condicionais têm por finalidade executar um ~~traco~~ trecho de código caso uma análise de um dado for verdadeiro, ou ainda, outro trecho, caso possuir valor lógico falso.

Os elementos são IF, ELSE IF, ELSE permitindo avaliar uma condição de determinado variável. Sintaxe, a condição e a estrutura de condição a ser verificada para decidir se as linhas do bloco de código serão executadas ou não.

if: é uma (bloco) estrutura condicional que executa a afirmação, dentro do bloco, se determinada condição for verdadeira.

else if: especifica uma nova condição para teste se o primeiro for falso.

else: se a mesma condição for falsa.