Introdução à Programação e Pensamento Computacional

Aula 1 – Pensamento Computacional

1 – Decomposição

O que é?

Primeiro passo da resolução de problemas dentro do conceito do PC (pensamento computacional)

Decompor é pegar um problema complexo e quebrá-lo em partes resolvíveis e fáceis de gerenciar

Estratégias:

Processo de quebrar e determinar partes menores e gerenciáveis – Análise; 1°

Combinar os elementos recompondo o problema original – Síntese; 2°

Ordem de execução de tarefas menores – Sequencial ou Paralela

Sequencial – Um problema depende do outro

Paralela - + Eficiência e – Tempo – Tarefas podem ser executadas concomitantemente

Decomposição – Treinamento

Maneiras distintas de decompor o mesmo problema

Como Decompor?

Identificar ou coletar dados

Agregar os dados

Funcionalidade

2 – Padrões

Modelo Base

Estrutura invariante

Repetição

Detectamos os padrões através das Similaridades e Diferenças

Por que determinar padrões?

Generalizar, com objetivo de obter resolução para problemas diferentes

Categorias – Dependem do Domínio

Classes

Pelo ser humano é feito através do Grau de Similaridade – Grupos conhecidos x Objetos desconhecidos

O computador reconhece padrões através da comparação

Representação de atributos para aprendizado, em seguida para armazenar dados e por último para regras de tomada de decisão

Aplicações: Classificação de dados, reconhecimento e imagem, reconhecimento de fala, análise de cenas, classificação de documentos

Áreas: Machine Learning, Redes Neurais, IA, Ciência de Dados

3 – Abstração

Generalização / Abstração

Abstrair – Observar, um ou mais elementos, avaliando características e propriedades em separado

Abstração Processo intelectual de isolamento de um objeto da realidade

Generalização na lógica é a operação intelectual que consiste em reunir uma classe geral, um conjunto de seres ou fenômenos similares

Como classificar os dados?

Características

Pontos essenciais Generalizar x Detalhar

4 – Algoritmos

Instruções detalhadas

O computador recebe, manipula e armazena dados

Processo de resolução de problemas passo a passo

Análise

Algoritmo

Codificação

Sequência de passos com objetivo definido

Execução de tarefas

Conjunto de operações que resultam em uma sucessão finita de ações

Como construir um algoritmo

Compreensão do problema

Definição de dados de entrada

Definir o processamento

Definir os dados de saída

Utilizar um método de construção

Teste e diagnóstico

Construção de Algoritmos

Narrativa

Fluxograma

Pseudocódigo

Aula 2 Introdução à Lógica de Programação

1 - O que é lógica?

Vem para resolver um problema;

Definição formal: Problema é uma questão que foge de uma determinada regra

Lógica é uma organização coesa, forma o raciocínio

É a organização e planejamento das instruções, assertivas em um algoritmo a fim de viabilizar uma implantação de um programa;

Seres humanos podem prever comportamentos, computadores não, por isso é usada a lógica em programação

2 – Técnicas de Lógica de Programação

Técnica Linear – Modelo Tradicional

Não tem vínculo – Execução Sequenciada

Recursos Limitados e uma única dimensão

Técnica Estruturada

Objetivo – Escrita de programas, entendimento, validação e manutenção (facilitador)

Pode ter mais de uma opção, ou seja, não é linear

Técnica Modular –

Partes independentes, onde temos regras independentes

Simplificação, decomposição do problema e verificação por módulo

Aula 3 – Fundamentos de Algoritmos

1 – Tipologia e Variáveis

Qual a função do Computador?

Processar as informações

Dados – Tratados e Processados –

Numéricos – Inteiros e Reais

Caracteres - Tamanho

Lógicos – True or False – Booleano

Instruções - Operações

Variável – Mutável, possui variações, incerto, inconstante e instável

Pode assumir qualquer um dos valores de um determinado conjunto de valores

Papéis de uma variável – Ação Modificação de estado

Controle Vigiada, controle

Constante é tudo aquilo que é fixo ou estável

2 – Instruções Primitivas

Determinam as ações que vamos executar, normalmente para cálculos matemáticos, por meio de operadores

Instruções – Palavras Chaves

Entrada, processamento e saída de dados

3 – Estruturas Condicionais e Operadores

Estado de uma pessoa ou coisa – Condição

Condição – Operação

Estruturas Condicionais – Simples, Compostas e encadeadas

Simples – Normal

Compostas – Exceções

Encadeadas – Sucessivas

Operadores Lógicos – And; Or; Not

Respostas simplificadas

4 – Estruturas de Repetição

Condições de Parada

Número de repetições pré-fixadas

Condições a serem satisfeitas

Vantagens – Redução de linhas, compreensão facilitada e redução de Erro

Enquanto (While)/ DoWhile (Repita)/ For (Para... De... Até)

5 – Vetores e Matrizes

Vetores tem um tamanho pré-fixado

Vetores – Unidimensionais

Matrizes - + de uma dimensão (coleção de vetores)

6 – Funções

Similar ao conceito de função na Matemática

Blocos de instruções (código) que realizam tarefas específicas, identificados por nomes e parâmetros

Decomposição do Algoritmo

Vantagens – Modularização do programa, código mais claro e conciso, reutilização de instruções

Definição, Nome Invocação e Variável local – São destruídas ao encerrar a função

Dados >>> Função >>> Resultado – Altera o estado do Programa

7 – Instruções de Entrada e Saída

Entrada – inserção de dados

Saída – Impressão de dados – Tela

Saída programada – Condicional ou Incondicional

Saída por interrupção – Algo deu errado

Saída

Bem-sucedida, erro de sintaxe, erro de programação, problemas com interface

Aula 4 – Linguagens de Programação

1 – Introdução a Linguagens de Programação

Breve história da computação

A evolução começa no hardware e vai para o software

Problemas computacionais –

problemas de decisão – Sim ou não – Ideia de Pertencimento

busca – Relacionamento Binário

otimização -maximizar ou minimizar uma otimização

O que é uma linguagem de programação – Método padronizado composto por um conjunto de regras sintáticas e semânticas de implementação de um código fonte

Conjunto de palavras e regras

Código fonte – Traduzido/Interpretado

2 – Como o computador entende o programa?

Um programa é um amontoado de palavras senão for possível que o computador entenda

Existe um processo de compilação – Compilador executa análise do programa e traduz para a linguagem de máquina

Tradução – Geração do programa objeto e Execução do programa objeto – Execução + rápida – Programas menores

Interpretação Programa fonte executado diretamente – Maior flexibilidade

3 – Características de um programa

Desenvolvimento de programas

Diretrizes:

Legibilidade – Estado que é legível

Redigibilidade – Facilidade de escrita de código – Coerência nas instruções

Confiabilidade - Faz o que foi programado para fazer

Custo – Análise de impacto

4 – Análises de Código

Análise Léxica – Particionar, Classificar e eliminar

Análise Sintática – Forma – Corretude do programa

Análise Semântica – Significado

5 – Paradigmas de Programação

Forma de resolver problemas limitada a um contexto específico

Paradigma Estruturado

Sequência

Decisão

Iteração

Ênfase em sequência

Utilização – Problemas Simples e diretos, aprender programação

POO ainda não compreendida por muitos

Orientação à objeto – baseado na utilização de objetos e suas interações

Atributos, métodos, estado

Pilares da orientação a objeto

Herança, encapsulamento, polimorfismo e abstração

POO – Reuso de código

PE – Problemas específicos diretos

Aula 5 – Algoritmos em portugol

Feito no portugol