Estrutura de Dados 1

Prof. Igor Calebe Zadi
igor.zadi@ifsp.edu.br

INSTITUTO FEDERAL
São Paulo
Campus Catanduva

I. Fundamentos de Estruturas de Dados

INSTITUTO FEDERAL
São Paulo
Campus Catanduva

I. Fundamentos de Estruturas de Dados

1. Definições
2. Classificação das Estruturas de Dados
3. Programação Orientada a Procedimentos
4. Tipos de dados primitivos
5. Cadeias de caracteres
6. Registros
7. Ponteiros

4. Tipos de Dados Primitivos

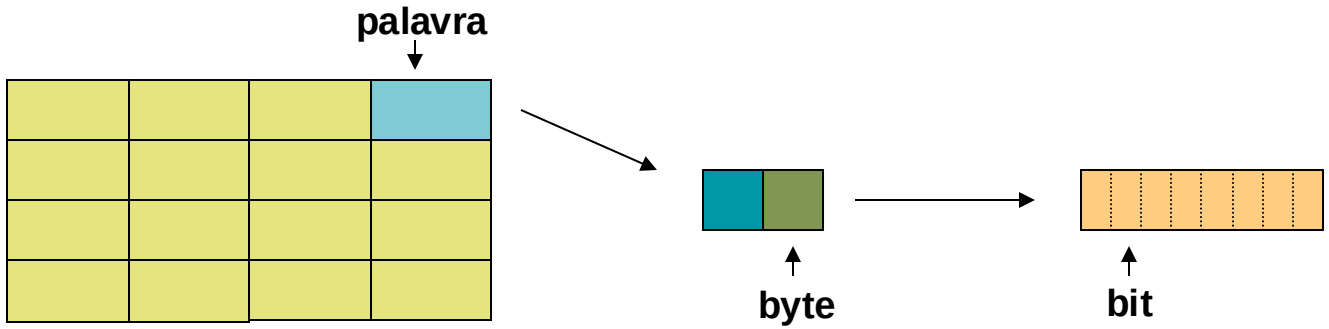
INSTITUTO FEDERAL
São Paulo
Campus Catanduva

4. Tipos de Dados Primitivos

- Tipos de dados (ED) fundamentais
- Usados para construção de outras ED
- São eles:
  - Inteiros
  - Reais
  - Lógicos
  - Caracteres

4. Tipos de Dados Primitivos

- Inteiros
  - podem ser positivos ou negativos
  - não possuem parte decimal
  - ocupam, em geral, uma palavra na memória
  - supondo uma palavra = 2 bytes
  - $2^8 \times 2^8 = 216 = 65536$ números diferentes



4. Tipos de Dados Primitivos

- Inteiros
  - É representado por uma sequência de números binários – para números positivos
  - Números negativos
    - Complemento de um - 0 para positivo, 1 para negativo (números invertidos)
    - Complemento de dois
      - Diferença em expressar quantidades positivas e negativas. Bit mais a esquerda é o mais significativo

4. Tipos de Dados Primitivos

- Inteiros – operações

Argumentos	Resultado	Operação
Par de Inteiros	Inteiro	soma (+), subtração (-), multiplicação (*), divisão inteira (DIV), resto da divisão (MOD)
Inteiro	Inteiro	oposto ou negação (-)
Par de Inteiros	Lógico	igualdade (=), diferença (!=), maior que (>), maior que ou igual a (>=), menor que (<), menor que ou igual a (<=)

4. Tipos de Dados Primitivos

- Reais
  - podem ser positivos ou negativos
  - possuem parte decimal (ponto flutuante)
  - ocupam, em geral, **duas palavras** na memória
    - supondo uma palavra = 2 bytes
    - $2^8 \times 2^8 \times 2^8 \times 2^8 = 2^{32}$ possibilidades de representação
  - Valores reais efetivamente representados, dependem da precisão e da quantidade de dígitos significativos

4. Tipos de Dados Primitivos

- Reais – operações

Argumentos	Resultado	Operação
Par de Reais	Real	soma (+), subtração (-), multiplicação (*), divisão (/)
Real	Real	oposto ou negação (-)
Par de Reais	Lógico	igualdade (==), diferença (!=), maior que (>), maior que ou igual a (≥), menor que (<), menor que ou igual a (≤)

4. Tipos de Dados Primitivos

- Lógicos
  - podem assumir os valores **verdadeiro** (*true*) ou **falso** (*false*)
  - por isso, são chamados de booleanos
  - **1 bit** seria suficiente para representar um tipo lógico, mas em geral ele ocupa **1 byte** ou, às vezes, **uma palavra** na memória

4. Tipos de Dados Primitivos

- Lógico – operações

Argumentos	Resultado	Operação
Par de Lógicos	Lógico	conjunção (AND), disjunção (OR), disjunção exclusiva(XOR)
Lógico	Lógico	negação (NOT)
Par de Lógicos	Lógico	igualdade (==), diferença (!=)

4. Tipos de Dados Primitivos

- Caracteres
  - são formados por um único caractere, tomado a partir do **alfabeto** válido
  - em geral, cada caractere ocupa **1 byte**
    - Existem $2^8 = 256$ caracteres possíveis, *a priori*

4. Tipos de Dados Primitivos

- Caracteres – operações

Argumentos	Resultado	Operação
Par de Caracteres	Lógico	igualdade (==), diferença (!=)
Caractere	Inteiro	posição do caractere no alfabeto
Par de caracteres	Lógico	Comparação das posições dos caracteres no alfabeto: maior que (>), maior que ou igual a (≥), menor que (<), menor que ou igual a (≤)

5. Cadeias de Caracteres

INSTITUTO FEDERAL
São Paulo
Campus Catanduva

5. Cadeias de Caracteres

- Cadeia ou *String* de caracteres
  - Sequência finita de símbolos tomados de um conjunto de caracteres (o **alfabeto**)
  - É uma **ED homogênea** construída a partir de uma ED primitiva, o caractere
  - Outros nomes:
    - sequências de caracteres
    - séries de caracteres

5. Cadeias de Caracteres

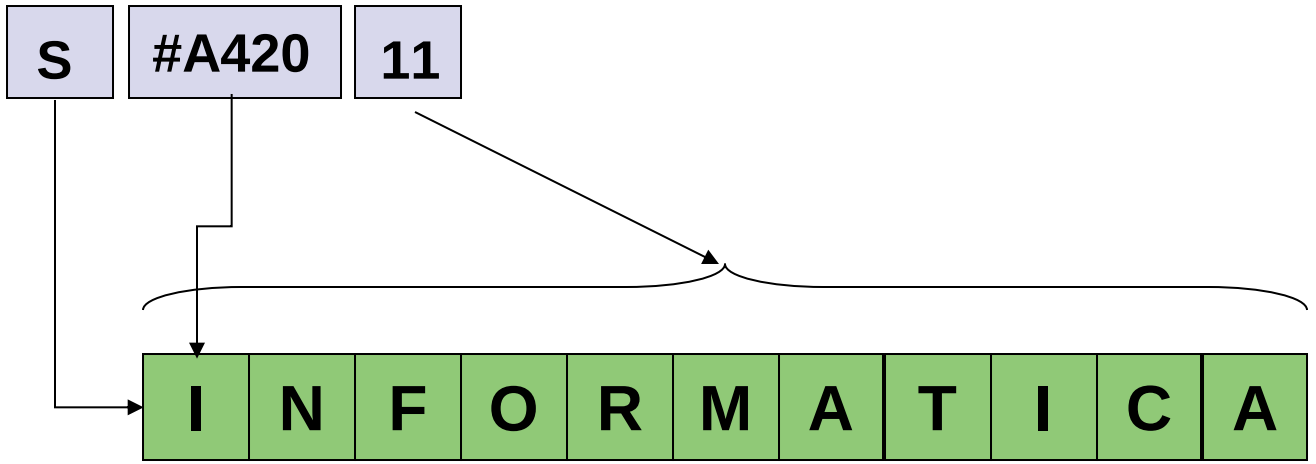
- Alfabeto *x* vocabulário
  - **Vocabulário**: conjunto de cadeias que de pode gerar a partir de um alfabeto
  - Exemplo:
    - alfabeto = { ‘C’, ‘D’, ‘1’ }
    - vocabulário = { “”, “1”, “CD1”, “DDC”, “1D111”, ... }
  - As cadeias são normalmente delimitadas por aspas duplas (“)
  - A **cadeia nula** ou vazia, indicada por “”, sempre faz parte de qualquer vocabulário, independente do alfabeto

5. Cadeias de Caracteres

- Controle do armazenamento
  - Existem basicamente três maneiras para controlar o armazenamento de uma **variável do tipo cadeia** na memória do computador:
    - nome da variável, endereço do início, tamanho
    - nome da variável, endereço do início, endereço do final
    - nome da variável, endereço do início, marca de final (caractere terminador)
  - Por este motivo não é possível, em geral, usar diretamente o operador de atribuição (=) para dar valor a uma variável do tipo cadeia

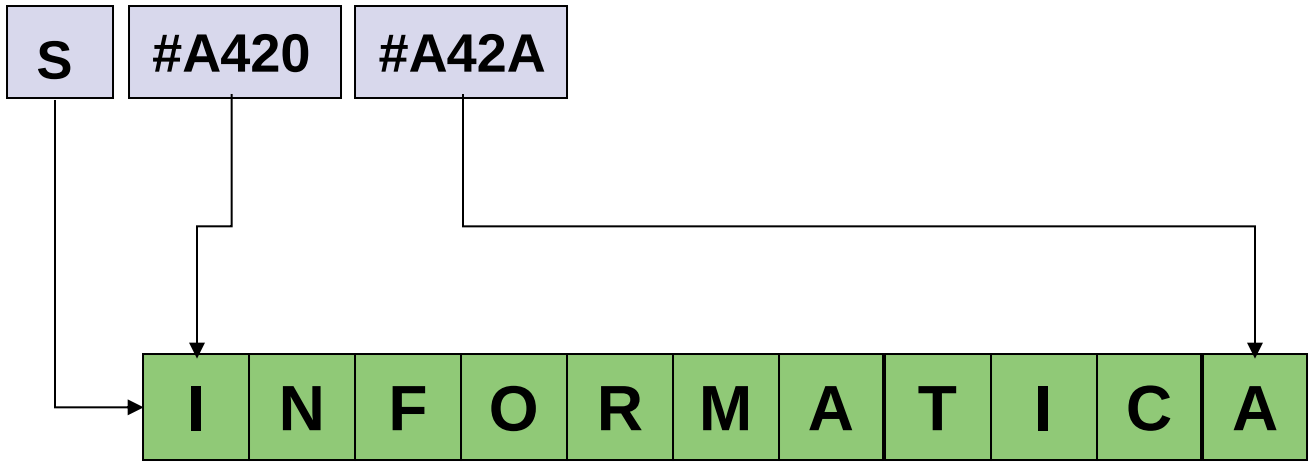
5. Cadeias de Caracteres

- Controle do armazenamento – opção 1
  - Nome da variável, endereço do início, tamanho



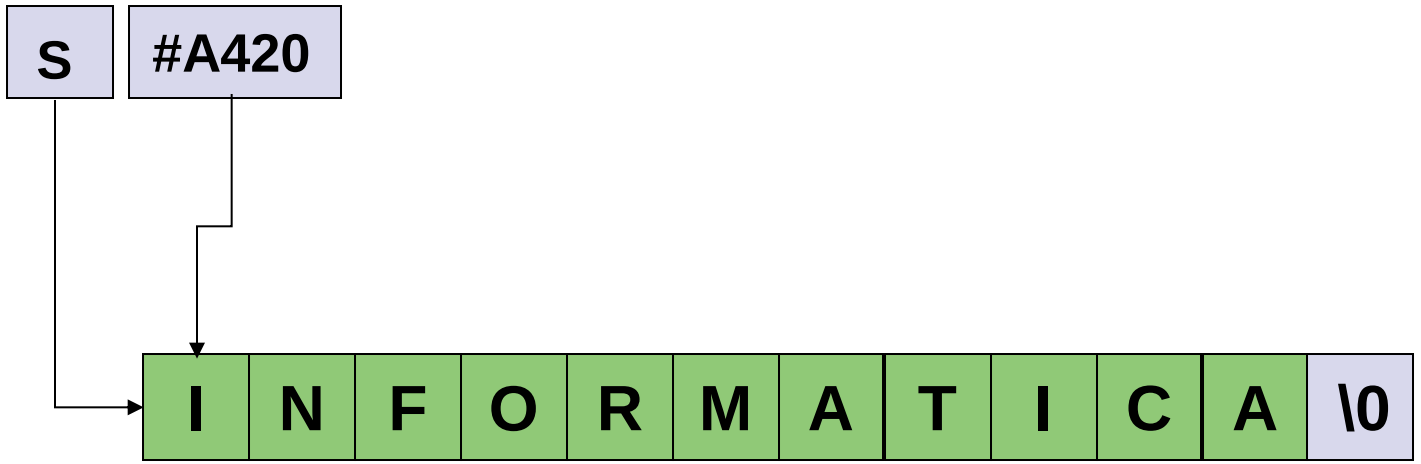
5. Cadeias de Caracteres

- Controle do armazenamento – opção 2
  - Nome da variável, endereço do início, endereço do final



5. Cadeias de Caracteres

- Controle do armazenamento – opção 3
  - Nome da variável, endereço do início, marca de final (caractere terminador)



5. Cadeias de Caracteres

- Operações com cadeias de caracteres
  - Três operações são **primitivas** (básicas)
    - PRIMEIRO (S_1)
    - RESTANTE (S_1)
    - CONCATENAÇÃO (S_1, S_2)
  - Todas as outras operações envolvendo cadeias podem ser escritas como combinação destas três operações primitivas
    - por isso, as outras são chamadas de **derivadas**

5. Cadeias de Caracteres

- PRIMEIRO (S_1)
  - Fornece como resultado o primeiro caractere da cadeia S_1
    - PRIMEIRO (“IGOR”) resulta no caractere ‘I’
  - Se a cadeia é nula
    - PRIMEIRO (“”) resulta na cadeia nula “”

5. Cadeias de Caracteres

- RESTANTE (S_1)
  - Fornece como resultado uma **nova** cadeia que é igual à própria cadeia S_1 após a remoção de seu primeiro caractere
    - RESTANTE (“IGOR”) resulta na cadeia “GOR”
  - Se a cadeia tem somente um caractere
    - RESTANTE (“A”) resulta na cadeia nula “”
  - Se a cadeia é nula
    - RESTANTE (“”) resulta na cadeia nula “”

5. Cadeias de Caracteres

- CONCATENAÇÃO (S_1 , S_2)
  - Fornece como resultado uma **nova** cadeia que é a junção das cadeias S_1 e S_2 (o final de S_1 é *emendado* com o início de S_2)
    - CONCATENAÇÃO (“Turma ”, “A”) resulta na cadeia “Turma A”
  - Na elaboração de algoritmos, é comum admitir que S_1 ou S_2 seja um caractere
    - CONCATENAÇÃO (“Professor”, ‘a’) resulta na cadeia “Professora”

5. Cadeias de Caracteres

- Operações com cadeias de caracteres
  - Existem diversas operações **derivadas**
    - COMPRIMENTO (S_1)
    - SUBCADEIA (S_1, i, n)
    - INSERÇÃO (S_1, S_2, i)
    - REMOÇÃO (S_1, i, n)
    - ...

5. Cadeias de Caracteres

- COMPRIMENTO (S_1)
  - Fornece como resultado um número **inteiro** correspondente à quantidade de caracteres da cadeia S_1
    - COMPRIMENTO (“AULA”) resulta no inteiro 4
  - Se a cadeia é nula
    - COMPRIMENTO (“”) resulta no inteiro zero
  - Se a cadeia tem caractere terminador, este não é contabilizado

5. Cadeias de Caracteres

- SUBCADEIA (S_1, i, n)
  - Fornece como resultado uma **nova** cadeia que é igual a parte da cadeia S_1
  - A cadeia-resultado tem *n* caracteres, contados a partir do *i*-ésimo caractere de S_1, inclusive
    - SUBCADEIA (“MARTE”, 2, 2) resulta na cadeia “AR”
    - SUBCADEIA (“MESA”, 4, 1) resulta na cadeia “A”
  - *i* e *n* são inteiros e, nesta notação:
    - $1 \leq i \leq$ COMPRIMENTO (S_1)
    - $1 \leq n \leq$ COMPRIMENTO (S_1)
    - $1 \leq (i + n - 1) \leq$ COMPRIMENTO (S_1)

5. Cadeias de Caracteres

- INSERÇÃO (S_1, S_2, i)
  - Fornece como resultado uma **nova** cadeia em que a cadeia S_2 é inserida na cadeia S_1 a partir de sua *i*-ésima posição, inclusive
    - INSERÇÃO (“ARA”, “EI”, 3) resulta na cadeia “AREIA”
    - INSERÇÃO (“BOFE”, “TE”, 5) resulta na cadeia “BOFETE”
  - *i* é inteiro e, nesta notação:
    - $1 \leq i \leq$ COMPRIMENTO (S_1) + 1

5. Cadeias de Caracteres

- REMOÇÃO (S_1, i, n)
  - Fornece como resultado uma **nova** cadeia que é cópia de S_1, exceto pela parte correspondente à SUBCADEIA (S_1, *i, n*), a qual foi suprimida (retirada)
    - REMOÇÃO (“DISCO”, 3, 2) resulta na cadeia “DIO”
    - REMOÇÃO (“PATA”, 4, 1) resulta na cadeia “PAT”
  - *i* e *n* são inteiros e, nesta notação:
    - $1 \leq i \leq$ COMPRIMENTO (S_1)
    - $1 \leq n \leq$ COMPRIMENTO (S_1)
    - $1 \leq (i + n - 1) \leq$ COMPRIMENTO (S_1)

5. Cadeias de Caracteres

- EXEMPLO
  - Sem utilizar a operação *Subcadeia*, escreve uma expressão (ou sequencia de expressões) que permite(m) obter a palavra “AJUSTAR” a partir das cadeias:
    - S1 = “ARBUSTO”
    - S2 = “DEVORAR”
    - S3 = “TRAJETO”

5. Exercício

- Sem utilizar a operação *Subcadeia*, escreva uma expressão (ou sequência de expressões) que permite(m) obter a palavra “COLHEITA” a partir das cadeias
  - S1= “CANETAS”
  - S2= “PINTADO”
  - S3= “FOLHETO”.

Observações sobre o material eletrônico

- O material ficará disponível na pasta compartilhada que é acessada sob convite
- O material foi elaborado a partir de diversas fontes (livros, internet, colegas, alunos etc.)
- Alguns trechos podem ter sido inteiramente transcritos a partir dessas fontes
- Outros trechos são de autoria própria
- Esta observação deve estar presente em qualquer utilização do material fora do ambiente de aulas do IFSP - Catanduva

INSTITUTO FEDERAL
São Paulo
Campus Catanduva