Tipos de dados, variáveis, constantes

Prof. Dr. Gerson Pastre de Oliveira

Computadores manipulam dados...

- Para processar respostas necessárias à solução de um problema ou tarefa, o computador tem um conjunto de hardware (equipamentos) e software (programas)
- O sistema operacional é um programa que interpreta as ações dos usuários, geralmente feitas por outros programas, e as submete aos equipamentos responsáveis por executá-las, em última instância
- Essas ações são geralmente constituídas a partir de dados, que precisam ser armazenados na memória do computador

Computadores manipulam dados...

- Não podemos esquecer que todo processamento parte do princípio que os dados manipulados estão circulando em memórias voláteis e são utilizados pelo processador
- Em um programa, os dados devem ser armazenados em espaços da memória reservados para esse fim pelo sistema operacional, que transforma a execução em um processo
- Esses espaços têm tamanho, nome e endereço, o que permite que sejam controlados, recuperados, reescritos e liberados sempre que necessário

Tipos de dados

- As linguagens de programação estruturadas geralmente suportam diferentes tipos de dados que podem ser utilizados para representar informações e manipulá-las em um programa
- **Inteiros (int)**: Podem ser positivos ou negativos, sempre sem casas decimais
- Podem ser usados para representar contagens, índices, dados como idade, quantidades de objetos não divisíveis etc.

Tipos de dados

- **Números de ponto flutuante (float/double):** os números de ponto flutuante representam números reais com casas decimais, significativas ou não
- Os tipos **float** e **double** são usados para representar números com precisão simples ou dupla, respectivamente
- Geralmente, representam medidas reais, valores monetários, distâncias não exatas, entre outras possibilidades

Tipos de dados

- **Caracteres (char):** os caracteres são usados para representar caracteres individuais, como letras, dígitos e símbolos
- São comumente usados para representar caracteres de texto em uma cadeia de caracteres (string)
- **Cadeias de caracteres (string):** são usadas para representar texto, como palavras e frases – são compostas por uma sequência de caracteres

Tipos de dados

- **Booleanos (bool):** representam um valor verdadeiro (true) ou falso (false)
- São comumente usados para representar condições lógicas em um programa
- Dependendo da linguagem de programação específica, podem haver outros tipos de dados disponíveis, como tipos personalizados, enumerações, entre outros

Capacidade de armazenamento

- **char:** 1 byte (8 bits)
- **short:** 2 bytes (16 bits)
- **int:** 4 bytes (32 bits)
- **long:** 8 bytes (64 bits)
- **float:** 4 bytes (32 bits)
- **double:** 8 bytes (64 bits)

Variáveis

- São espaços reservados na memória do computador e usados para armazenar dados (apenas um, que pode ser sobrescrito)
- As variáveis têm um nome que é usado para referenciar seu valor no código-fonte do programa (**identificadores**)
- De maneira geral, as variáveis são usadas para armazenar temporariamente valores que são manipulados e processados pelo programa

Variáveis

- Geralmente, as variáveis são declaradas em um programa com um tipo de dado específico, como inteiro, ponto flutuante, caractere, booleano, entre outros
- O tipo de dado determina o tamanho e o formato dos valores que podem ser armazenados na variável
- Para usar uma variável em um programa, é necessário inicializá-la com um valor, que pode ser inserido por meio de uma **operação de atribuição** ou por **entrada de dados**

Constantes

- São valores fixos que são definidos no código-fonte do programa e que não podem ser alterados durante a execução do programa (também usam a memória disponível)
- Ou seja, ao contrário das variáveis, que podem ter seu valor modificado durante a execução do programa, as constantes têm um valor definido e imutável

Constantes

- Constantes são usadas para representar valores que são conhecidos e não mudam durante a execução do programa, como o número de dias em uma semana ou o valor de pi, por exemplo
- Assim com as variáveis, constantes geralmente são declaradas em um programa com um tipo de dado específico, como inteiro, ponto flutuante, caractere, booleano, entre outros

Identificadores

- Em C, um identificador é um nome dado a uma variável, função, estrutura, constante ou qualquer outro elemento do programa
- Não existem “nomes” (identificadores) obrigatórios e tampouco a linguagem de programação interpreta sua coerência – entretanto, a ideia é usar bom-senso
- Algumas boas práticas para criar identificadores em C incluem escolher nomes que sejam descritivos e fáceis de entender, evitando abreviações desnecessárias e nomes muito longos, e usando um padrão consistente de nomenclatura em todo o código

Regras para identificadores

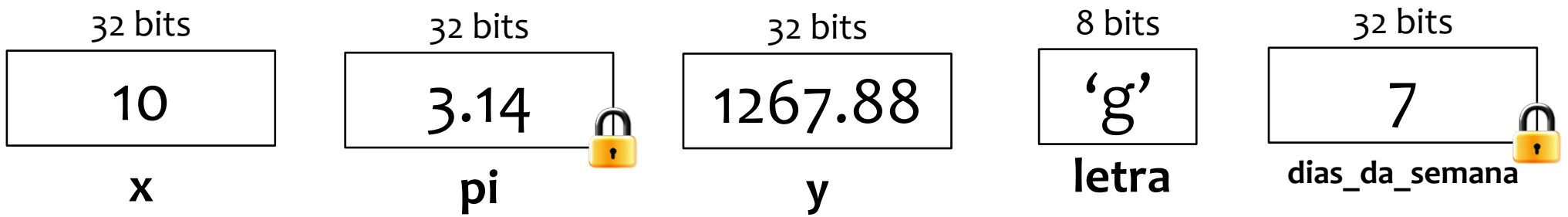
- Os identificadores podem conter letras maiúsculas e minúsculas, dígitos numéricos e o caractere de sublinhado "_" (não podem conter caracteres especiais como #, $, %, entre outros)
- O primeiro caractere do identificador deve ser uma letra ou um sublinhado "_" (números não são permitidos como primeiro caractere)
- Os identificadores são *case-sensitive*, ou seja, diferenciam letras maiúsculas e minúsculas – por exemplo, o identificador "numero" é diferente de "Numero"

Regras para identificadores

- Os identificadores não podem ser palavras reservadas da linguagem, como int, char, if, else ou for, por exemplo – essas palavras são usadas pela linguagem de programação para construir a estrutura do código e, portanto, não podem ser usadas como identificadores
- O comprimento máximo de um identificador varia de acordo com a implementação da linguagem (em geral, a maioria dos compiladores C permite identificadores com até 31 caracteres)

Exemplo declaração de variáveis e constantes

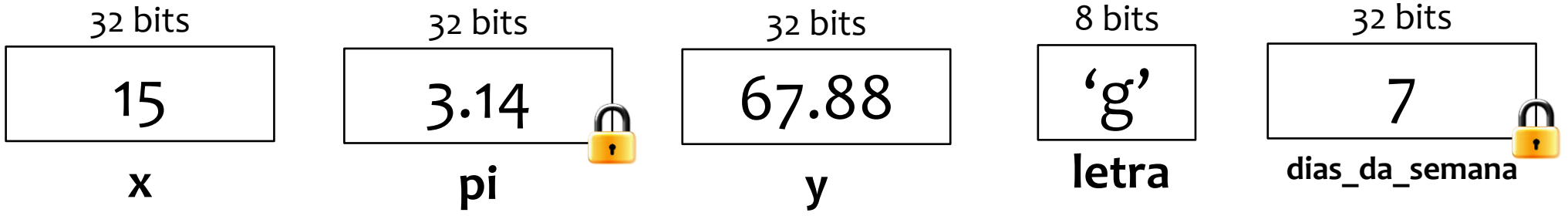
```c
#include<stdio.h>
main()
{
    int x; // declaração de uma variável inteira
    float y; // declaração de uma variável de ponto flutuante de precisão simples
    char letra; // declaração de uma variável caractere (armazena UM caractere)
    const float pi = 3.14; // declaração de uma constante de ponto flutuante
    const int dias_da_semana = 7; // declaração de uma constante inteira
    x = 10;
    y = 1267.88;
    letra = 'g';
    const int k = x + 20;
}
```



Observações:

x = x + 5;

y = y – 1200;



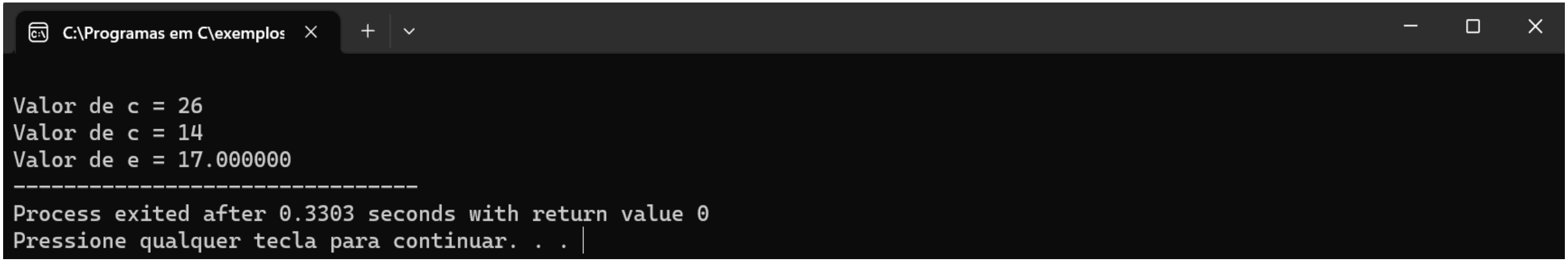
Operador de atribuição

- Nos exemplos anteriores, o sinal de igualdade (=) apareceu quando se quis inserir (atribuir) um valor relacionado a uma variável
- O sinal de igualdade é, portanto, o **operador de atribuição**, cuja função é fazer com que os dados sejam armazenados em uma variável ou em uma constante, por meio da imposição de um valor diretamente ou usando de outras operações, como as de natureza aritmética

Operadores aritméticos

Operador	Símbolo
Adição	+
Subtração	–
Multiplicação	*
Divisão	/
Resto de divisão	%

```c
#include<stdio.h>
main()
{
    int a, b, c;
    a = 20;
    b = 6;
    c = a + b; // o valor de c é 26
    printf ("\nValor de c = %d",c); //saída de dados
    c = a - b; // o valor de c é sobrescrito e passa a ser 14
    printf ("\nValor de c = %d",c);
    float d, e;
    d = 8.5;
    e = d * 2;
    printf ("\nValor de e = %f",e);
}
```



```c
#include<stdio.h>
main()
{
    int a, b, c;
    a = 20;
    b = 4;
    c = a / b; // o valor de c é 5 (o quociente da divisão de 20 por 4)
    printf ("\nValor de c = %d",c);
    b = b + 2; // o valor de b é sobrescrito e passa a ser 6
    c = a % b; // o valor de c é 2 (o resto da divisão de 20 por 6)
    printf ("\nValor de c = %d",c);
}
```

```
C:\Programas em C\exemplos

Valor de c = 5
Valor de c = 2
--------------------------------
Process exited after 0.2347 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```