

Tipos de dados Armazenamento

Prof. Tiago Massoni
Prof. Fernando Buarque

Engenharia da Computação

Poli - UPE

Informações

- Computador: máquina que manipula informações (dados)
 - Dado vs informação
- A unidade básica da informação é o **bit**
 - Cada um: duas possibilidades
- n chaves representam 2^n valores possíveis

2

Tipos de dados

- Objetos do mundo real representam algum tipo específico
 - Números, letras, frações
- Computadores processam dados conforme seu tipo
 - Operadores e operandos
 - Processamento e dados

3

Tipos de dados

- Primitivos (básicos)
 - Inteiros
 - Reais
 - Caracteres
- Compostos (agregados)
 - Estruturas (objetos)
 - Strings
 - Arrays (Vetores)

4

Inteiros

- Representação de números inteiros: **decimal**
- Representação em bits: **binário**
- Binário simples
 - $10 = 2$ ($0 \cdot 1 + 1 \cdot 2$)
 - $100110 = 38$ ($1 \cdot 0 + 2 \cdot 1 + 4 \cdot 1 + 8 \cdot 0 + 16 \cdot 0 + 32 \cdot 1$)
- toda string de bits de tamanho n representa um inteiro não-negativo único, entre 0 e $2^n - 1$
- E números negativos?

5

Inteiros negativos

- Dois métodos podem ser usados
- Complemento de 1
 - número negativo é representado mudando cada bit para a definição do bit oposto
 - Ex: 00100110 representa 38, 11011001 é usado para representar -38
- Complemento de 2 (**C, Java**)
 - 1 é somado à representação de complemento de um de um número negativo
 - Ex: 11011010 é usado para representar -38

6

Reais

- Notação ponto flutuante
 - Mantissa x base elevada a uma potência de inteiro
 - Ex: $387,53 (= 38753 * 10^{-2})$
- Em computadores digitais de 32 bits de endereçamento tem mantissa 24 bits, 8 para expoente e base fixa 10
 - Ex: a representação binária de 24 bits de 38753 é 000000001001011101100001,
 - Representação binária de complemento de dois de oito bits de -2 é 11111110
 - Representação de 387,53 é 00000000100101110110000111111110
- Vantagem: números muito grandes

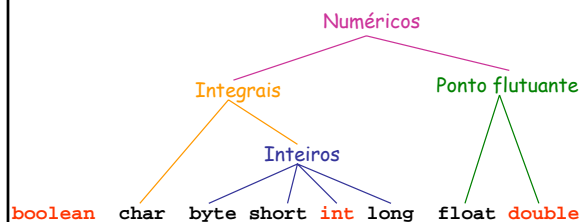
7

Caracteres

- Representação de valores não-numéricos (letras)
- Número pré-determinado de bits determinam um código (ASCII, Unicode)
 - 8 bits (byte)
 - 16 bits (2 bytes)

8

Os tipos primitivos de Java



9

Tipos de Dados Primitivos

Tipo	Descrição	Tamanho
(inteiros)		
byte	Inteiro de 8 bits	1 byte
short	Inteiro de 16 bits	2 bytes
int	Inteiro de 32 bits	4 bytes
long	Inteiro de 64 bits	8 bytes
(Números Reais)		
float	Ponto flutuante precisão simples	4 bytes IEEE 754
double	Ponto flutuante precisão dupla	8 bytes IEEE 754
(Outros Tipos)		
char	Caracter	16 bits - Unicode
boolean	true ou false	true ou false (1 bit)

10

Valores dos tipos: exemplos

Tipo	Exemplo
(inteiros)	
byte	100
short	350
int	2873, 034, 0xF
long	8864L
(Números Reais)	
float	23.746F
double	23.746
(Outros Tipos)	
char	'c', '\u4567'
boolean	true, false

11

Variáveis

- Guarda um valor
 - de um tipo específico
 - Declaração deve possuir um nome e um tipo
- Será reservado espaço uma posição de memória referenciada pelo identificador **a**
- Um identificador é usado no lugar de um endereço numérico para citar determinada posição de memória
 - conveniente para o programador
- Conteúdo interpretado como inteiro (próximos 32 bits)

int a;

tipo → int
identificador → a

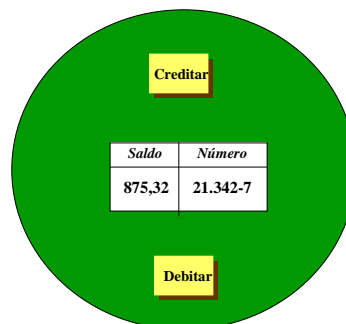
12

Java é orientada a objetos

- Uso de abstrações bem mais próximas do mundo do problema
 - Objetos, não funções
- Em um programa, "tudo" é objeto
- Um programa é um monte de objetos dizendo aos outros o que fazer
 - Mensagens

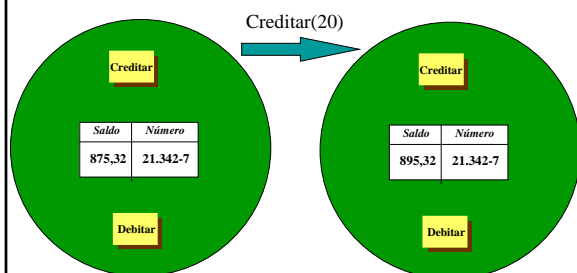
13

Objeto Conta Bancária



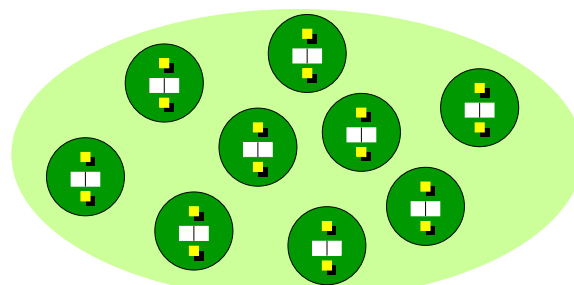
14

Estados do Objeto Conta



15

Classe de Contas Bancárias



Objetos semelhantes moldados por uma classes

16

Atributos em Java (estrutura)

```
public class Conta {  
    private int numero;  
    private double saldo;  
    ...  
}
```

- Cada atributo tem um tipo específico que caracteriza as propriedades dos objetos da classe
- int e double denotam os tipos cujos elementos são inteiros e strings
- Atributos privados

17

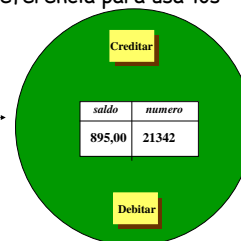
Objetos e Classes

- Programa é uma sequência de operações em objetos de várias classes
 - Objetos: área de memória
 - Precisamos do tipo referência para usá-los

Variável do tipo Conta

c

c.numero = 21342;
c.saldo = 895.0;



18

Strings

- Sequências de caracteres
- Não existe tipo primitivo string em Java
- API de Java possui classe chamada String

Cria um novo objeto do tipo String e o armazena ref. em saudacoes

```
String saudacoes = "Ola";  
String saudacoes = new String("Ola");
```

19

Nova classe Conta

```
public class Conta {  
    private String numero;  
    private double saldo;  
    ...  
}
```

Números podem conter traços, barras

20

Comparação de Strings

```
String saudacoes = "Ola";  
String cumprimento = new String("Ola");  
if (saudacoes == cumprimento)  
    //E agora?
```



21

Comparação de Strings

- Comparar variáveis String não tem resultado desejado (referências)
- Usar método equals() da classe String

```
String saudacoes = "Ola";  
String cumprimento = new String("Ola");  
if (saudacoes.equals(cumprimento))  
    //Agora sim!!
```

22

Comparação e tamanho

- boolean equals(umString)
- boolean equalsIgnoreCase(umString)
- int length()

```
String a = "Sharon Stone";  
String b = "sharon stone";  
int comprimento = a.length();  
boolean resposta1 = a.equals(b);  
boolean resposta2 = a.equalsIgnoreCase(b);  
boolean resposta3 = b.equalsIgnoreCase(a);
```

Qual é o valor de cada resposta?

23

StringBuffer

- Podem ser manipuladas (mutáveis)

```
StringBuffer dest = new StringBuffer("Hello!");  
dest.insert(0, "Hey ");  
System.out.println(dest);
```

Hey Hello!

24

Arrays (Vetores)

- São objetos especiais de Java
- Armazenam dados de um determinado tipo (homogêneo)
- `Tipo[]` é a classe
- Cada componente é identificado por um índice
- O primeiro elemento do array tem índice 0 e o último tem índice tamanho - 1

25

Declaração, criação, inicialização

```
int[] a;  
double[] x;  
  
a = new int[100];  
x = {10.0, 15.0, 20.0};  
  
double[] salarios = new double[5];  
  
for (int i = 0; i < salarios.length; i++) {  
    salarios[i] = i * 1000;  
}
```

Forma mais direta de inicialização

Inicialização com laço

26

Acesso a elementos do array

`variável[expressão_inteira]`

- Escrita e leitura a elementos do array é feito através de índices

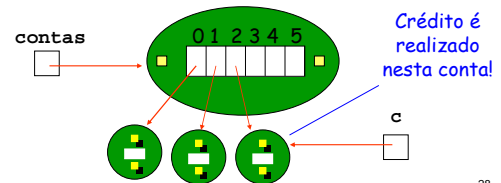
```
double [] salarios = {20.0,30.0,50.0};  
salarios[0] = salarios[0] * 1000;  
System.out.println(salarios[2]);
```

27

Arrays de objetos

- Aliasing garante modificação dos objetos do array

```
Conta c = contas[2];  
c.creditar(20);
```



28

Estruturas de dados

- Estudo tem duas metas
 - Identificar entidades e operações para resolver problemas
 - Determinar representações concretas para entidades abstratas do mundo real
- Estruturas de dados para resolver problemas - montagem de estruturas usando estruturas mais simples
- Motivações
 - Utilidade
 - Raciocínio

29