





Sun Certified Programmer for the Java 2 Platform

Fernando Anselmo

fernando.anselmo@dfjug.org

Programa

dia 1 – Fundamentos da Linguagem e Garbage Collection

dia 2 – Operadores e Assinaladores

dia 3 – Declarações e Controle de Acesso

dia 4 – Conversão e Casting

dia 5 – Fluxo de Controle, Exception e Assertions

dia 6 – Overloading, Override, Tipo de Runtime e OO

dia 7 – Threads

dia 8 – Classes fundamentais (Math, String, Wrapper)

dia 9 – Object e Coleções

dia 10 – Questões “Bizarras”

Sobre a Prova

- Informações na Página:

<http://suned.sun.com/US/certification/java/index.html>

- **Prometric:** <http://www.prometric.com> (Schedule A Test)
- Sun Certified Programmer for the Java 2 Platform 1.4 e 5.0

CX-310-035 - CX-310-055

- Total de questões: 61 e 72 (subjetiva e objetiva)
- Para passar: 52% (32 corretas) e 59% (43 corretas)
- Tempo máximo: 120 minutos e 175 minutos
- Telefone da Sun Education: **0800 55 7863**
- Valor do voucher: **330,00 reais**

Dia 1

Language Fundamentals

- Identificar corretamente as construções;
- Conhecer as palavras chaves e os identificadores;
- Intervalo de todos os tipos primitivos;
- Escrever declarações de classes e métodos;
- Trabalhar corretamente com arrays;
- Determinar efeitos sobre os objetos e valores primitivos através da passagem de parâmetros; e
- Entender o funcionamento correto do “Garbage Collector”.

Source Files

- Todo arquivo possui a extensão .java;
- Contém uma classe definida como pública (com nome idêntico ao arquivo) e ilimitadas classes particulares;
- A classe se apresenta na seguinte ordem: Declaração package, Declarações import e Declarações da Classe; e
- Para o arquivo: Test.java

```
package exame.guia;  
import java.util.*;  
public class Teste { ... }  
class Teste2 { ... }
```

Keywords and Identifiers

- Lista das Palavras chaves e reservadas*:

abstract	assert	boolean	break	byte	case
catch	char	class	<i>const*</i>	continue	default
do	double	else	extends	false	for
final	finally	float	<i>goto*</i>	<i>if</i>	<i>int</i>
<i>import</i>	<i>implements</i>	<i>instanceof</i>	<i>interface</i>	<i>long</i>	<i>native</i>
<i>new</i>	<i>package</i>	<i>private</i>	<i>protected</i>	<i>public</i>	<i>return</i>
<i>short</i>	<i>static</i>	<i>strictfp</i>	<i>super</i>	<i>switch</i>	<i>enum</i>
<i>synchronized</i>	<i>this</i>	<i>throw</i>	<i>throws</i>	<i>transient</i>	<i>true</i>
<i>try</i>	<i>void</i>	<i>volatile</i>	<i>null</i>	<i>while</i>	

- Nomes iniciam por letras (a..z A..Z), sinal de dolar (\$) ou underscore (_) depois é permitido dígitos (0..9)

Primitive Data Types

- Os tipos primitivos são:

boolean (1 bit true ou false)

byte (8 bits -2^7 a 2^7-1)

int (32 bits -2^{31} a $2^{31}-1$)

float (32 bits -10^{38} a 10^{38})

char (16 bits 0 a $2^{16}-1$)

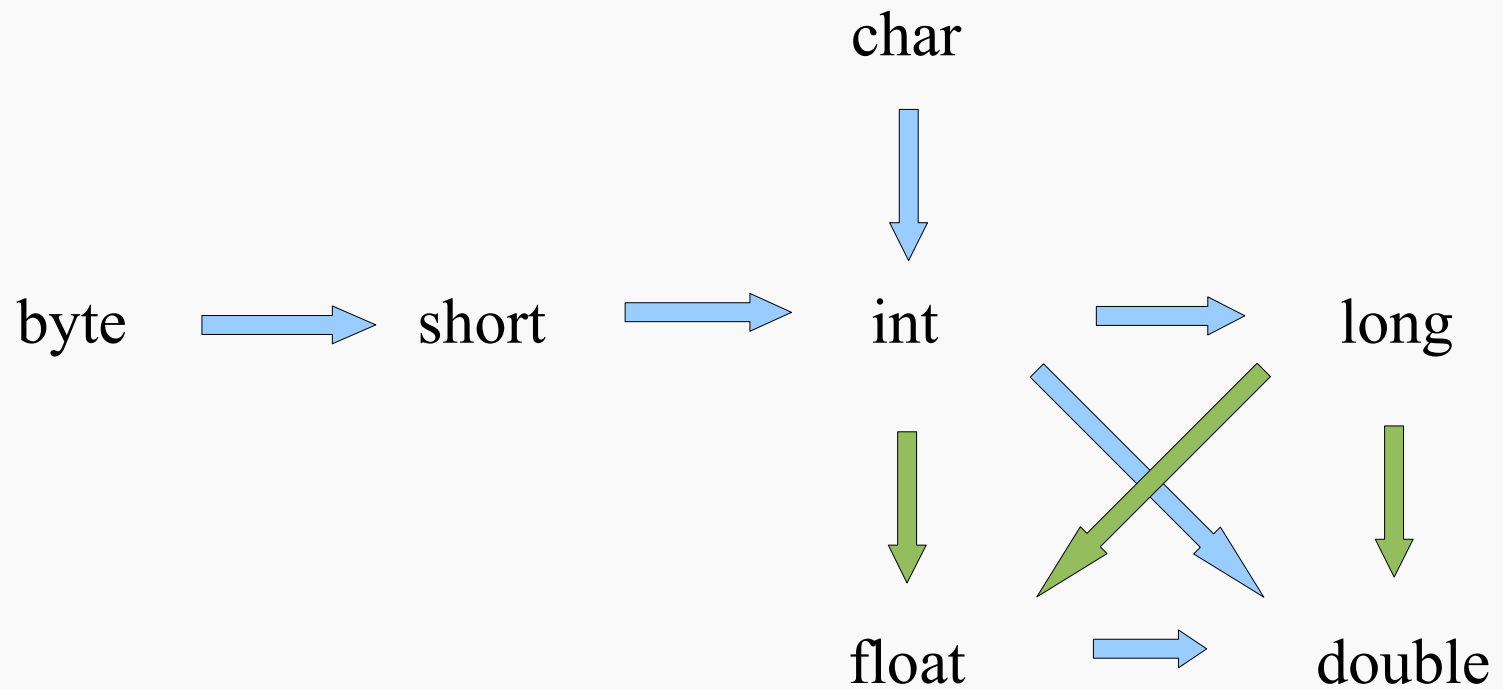
short (16 bits -2^{15} a $2^{15}-1$)

long (64 bits -2^{63} a $2^{63}-1$)

double (64 bits -10^{308} a 10^{308})

- Float (7 dig) e double (15 dig) são tipo IEEE 754 variando de: NaN – Not a Number, NEGATIVE_INFINITY a POSITIVE_INFINITY; e
- O sufixo F ou f (1.82f) indica float literal, D ou d indica double literal (1.82d) e um L ou l indica um long (12L).

Conversão



Array

- Para criar um array é preciso: Declarar, Construir e Inicializar;
- Cada conjunto de colchetes define uma dimensão do array;
- Valores iniciais dos tipos primitivos:

`boolean (false)`

`byte (0)`

`int (0)`

`float (0.0f)`

`char ('\u0000')`

`short (0)`

`long (0L)`

`double (0.0d)`

- Declarado de objeto será inicializado com “null”; e
- Índice do array é iniciado com [0] e o atributo length possui seu tamanho.

Class Fundamentals

- O método main é o ponto de entrada da aplicação:

```
public static void main(String[] args)
```
- Ele é declarado público por convenção e static por necessidade;
- Recebe parâmetros através da chamada a classe:

```
java Países Brasil EUA
```
- `args[0]` terá “Brasil” e `args[1]` terá “EUA”.

Argument Passing

- Um método é definido por:

```
modificador retorno nome([parametro]) {...}
```

- Pode-se passar qualquer tipo primitivo ou objeto por parâmetro a um método, sendo os primeiros são “cópias” e não atributos originais;
- Objetos são modificados por parâmetros (exceto obj String por característica de imutabilidade);
- No caso do método não retornar nada, é utilizada a palavra-chave “void”; e
- Tipos de Modificadores:
 - `public` - Sem restrições de acesso
 - `protected` - Com permissão ao mesmo package
 - `private` - A própria classe que o definiu

Garbage Collector

- Não é possível a destruição explícita dos objetos;
- Ativado automaticamente na existência de falta de memória;
- Tem baixa prioridade;
- O método `finalize()` do objeto é invocado antes que o `gc` seja executado;
- Os métodos `System.gc()` e `Runtime.gc()` sugerem que a JVM faça uma reciclagem dos objetos não utilizados; e
- O `gc` não previne memória baixa.

Time for Exercises

