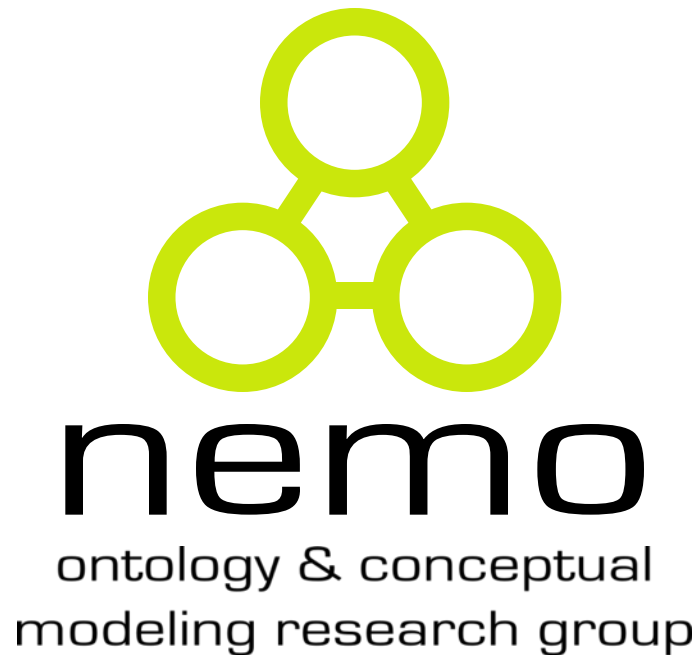




Linguagens de Programação

10 – Avaliação de Linguagens

Vítor E. Silva Souza

(vitorsouza@inf.ufes.br)

<http://www.inf.ufes.br/~vitorsouza>

Departamento de Informática

Centro Tecnológico

Universidade Federal do Espírito Santo



Esta obra foi licenciada sob uma Licença [Creative Commons Atribuição 3.0 Não Adaptada](https://creativecommons.org/licenses/by-sa/3.0/).

- Introdução;
 - Amarrações;
 - Valores e tipos de dados;
 - Variáveis e constantes;
 - Expressões e comandos;
 - Modularização;
 - Polimorfismo;
 - Exceções;
 - Concorrência;
 - ➔ Avaliação de linguagens.
-

- Estes slides foram baseados em:
 - Slides do prof. Flávio M. Varejão;
 - Livro “Linguagens de Programação – Conceitos e Técnicas” (Varejão);
 - Livro “Linguagens de Programação – Princípios e Paradigmas, 2a edição” (Tucker & Noonan).

- Fundamentais para o profissional de computação;
- Envolvida em diversas etapas do processo de software:
 - Viabilidade de um projeto;
 - Estimativa de tempo e custo;
 - Definição do paradigma de desenvolvimento; etc.
- Conhecimento dos conceitos teóricos:
 - Maior habilidade para resolver problemas;
 - Mais facilidade para aprender novas LPs;
 - Maior capacidade de escolher a LP para um projeto.

- Dificuldade para definir:
 - Inúmeras possibilidades;
 - Granularidade;
 - Dependência do contexto;
 - Opiniões divergentes;
 - Etc.

Isso se aplica mesmo a essa aula (visão do professor sobre o assunto) e ao capítulo do livro do Flávio (visão do autor sobre o assunto)!

- Aplicabilidade;
- Confiabilidade;
- Facilidade de aprendizado;
- Eficiência;
- Portabilidade;
- Suporte ao método de projeto;
- Evolutibilidade;
- Reusabilidade;
- Integração com outros softwares;
- Custo.

- Escopo;
- Expressões e comandos;
- Tipos primitivos e compostos;
- Gerenciamento de memória;
- Persistência de dados;
- Passagem de parâmetros;
- Encapsulamento e proteção;
- Sistema de tipos;
- Verificação de tipos;
- Polimorfismo;
- Exceções;
- Concorrência.

- Dificuldade para realizar;
- Não existe uma LP melhor que outra:
 - Depende do contexto de uso;
 - Depende do critério avaliado;
- Comparações refletem visão do autor.

Comparação entre C, C++ e Java

Crítérios gerais	C	C++	Java
Aplicabilidade	Sim	Sim	Parcial
Confiabilidade	Não	Não	Sim
Aprendizado	Não	Não	Não
Eficiência	Sim	Sim	Parcial
Portabilidade	Não	Não	Sim
Método de projeto	Estruturado	Estruturado e OO	OO
Evolutibilidade	Não	Parcial	Sim
Reusabilidade	Parcial	Sim	Sim
Integração	Sim	Sim	Parcial
Custo	Depende da ferramenta	Depende da ferramenta	Depende da ferramenta

Comparação entre C, C++ e Java

Crítérios gerais	C	C++	Java
Aplicabilidade	Sim	Sim	Parcial
Confiabilidade	C e C++ são linguagens de propósito geral. Java, contudo, não oferece recursos para controlar diretamente o hardware, obrigando o programador a usar métodos nativos.		
Aprendizado			
Eficiência			
Portabilidade			
Método de projeto		OO	
Evolutibilidade	Não	Parcial	Sim
Reusabilidade	Parcial	Sim	Sim
Integração	Sim	Sim	Parcial
Custo	Depende da ferramenta	Depende da ferramenta	Depende da ferramenta

Comparação entre C, C++ e Java

CrITÉrios gerais	C	C++	Java
Aplicabilidade	Sim	Sim	Parcial
Confiabilidade	Não	Não	Sim
Aprendizado	C e C++ possuem inúmeras características estimuladoras de erros em programação (ex.: desvio incondicional irrestrito, aritmética de ponteiros, etc.). Java centraliza certas operações para evitar problemas (ex.: verificação de índices de vetor, coleta de lixo, etc.)		
Eficiência			
Portabilidade			
Método de projeto			
Evolutibilidade			
Reusabilidade	Parcial	Sim	Sim
Integração	Sim	Sim	Parcial
Custo	Depende da ferramenta	Depende da ferramenta	Depende da ferramenta

Comparação entre C, C++ e Java

Critérios gerais	C	C++	Java
Aplicabilidade	Sim	Sim	Parcial
Confiabilidade	Não	Não	Sim
Aprendizado	Não	Não!	Não
Eficiência	Embora C e Java sejam mais fáceis de aprender que C++, nenhuma das LPs atende ao critério. C exige uso massivo de ponteiros, que não é um conceito trivial. Java apresenta muitos conceitos, nem sempre ortogonais. Em especial as versões mais recentes adicionaram bastante complexidade (ex.: tipos genéricos, closures, etc.).		
Portabilidade			
Método de projeto			
Evolutibilidade			
Reusabilidade			
Integração			
Custo	Depende da ferramenta	Depende da ferramenta	Depende da ferramenta

Comparação entre C, C++ e Java

Critérios gerais	C	C++	Java
Aplicabilidade	Sim	Sim	Parcial
Confiabilidade	Não	Não	Sim
Aprendizado	Não	Não	Não
Eficiência	Sim	Sim	Parcial
Portabilidade	O critério toma como base programadores muito experientes. Neste caso, C permite um controle mais fino e é portanto mais eficiente. C++ vai na mesma linha. Java assume o controle de diversos aspectos, adicionando mecanismos de verificação, coleta de lixo, etc. que diminuem a eficiência.		
Método de projeto			
Evolutibilidade			
Reusabilidade			
Integração			
Custo	Depende da ferramenta	Depende da ferramenta	Depende da ferramenta

Comparação entre C, C++ e Java

Crítérios gerais	C	C++	Java
Aplicabilidade	Sim	Sim	Parcial
Confiabilidade	Não	Não	Sim
Aprendizado	Não	Não	Não
Eficiência	Sim	Sim	Parcial
Portabilidade	Não	Não	Sim
Método de projeto	Embora C e C++ sejam padronizadas pela ANSI, é comum compiladores diferentes terem características diferentes. Java, por outro lado, tem a portabilidade como uma característica fundamental da LP.		
Evolutibilidade			
Reusabilidade			
Integração			
Custo	Depende da ferramenta	Depende da ferramenta	Depende da ferramenta

Comparação entre C, C++ e Java

Crítérios gerais	C	C++	Java
Aplicabilidade	Sim	Sim	Parcial
Confiabilidade	Não	Não	Sim
Aprendizado	Não	Não	Não
Eficiência	Sim	Sim	Parcial
Portabilidade	Não	Não	Sim
Método de projeto	Estruturado	Estruturado e OO	OO
Evolutibilidade	Esse critério depende da escolha do paradigma adotado no projeto. Contudo, cada vez mais o paradigma OO vem sendo adotado em projetos de sistemas de informação.		
Reusabilidade			
Integração			
Custo			
	ferramenta	ferramenta	ferramenta

Comparação entre C, C++ e Java

Crítérios gerais	C	C++	Java
Aplicabilidade	Sim	Sim	Parcial
Confiabilidade	Não	Não	Sim
Aprendizado	Não	Não	Não
Eficiência	Sim	Sim	Parcial
Portabilidade	Não	Não	Sim
Método de projeto	Estruturado	Estruturado e OO	OO
Evolutibilidade	Não	Parcial	Sim
Reusabilidade	C possui características que permitem código ilegível e difícil de manter. C++ melhora esse aspecto quando a orientação a objetos é usada (estímulo ao encapsulamento e abstração). Java só admite programação OO e ainda oferece estímulo para construção de código bem documentado (ex.: JavaDoc).		
Integração			
Custo			

Comparação entre C, C++ e Java

Crítérios gerais	C	C++	Java
Aplicabilidade	Sim	Sim	Parcial
Confiabilidade	Não	Não	Sim
Aprendizado	Não	Não	Não
Eficiência	C oferece apenas reuso de funções, tipos e variáveis distribuídas em bibliotecas. C++ e Java oferecem o conceito de classes e possuem mecanismo de pacotes. O polimorfismo universal também auxilia na criação de código reutilizável e <i>frameworks</i>.		
Portabilidade			
Método de projeto			
Evolutibilidade			
Reusabilidade	Parcial	Sim	Sim
Integração	Sim	Sim	Parcial
Custo	Depende da ferramenta	Depende da ferramenta	Depende da ferramenta

Comparação entre C, C++ e Java

CrITÉrios gerais	C	C++	Java
Aplicabilidade	Sim	Sim	Parcial
Confiabilidade	Não	Não	Sim
Aprendizado	Não	Não	Não
Eficiência	Sim	Sim	Parcial
Portabilidade	Não	Não	Sim
Método de projeto	C e C++ podem invocar código compilado por qualquer LP. Java tem que recorrer ao mecanismo JNI (Java Native Interface), que integra com C/C++ apenas.		
Evolutibilidade			
Reusabilidade			
Integração	Sim	Sim	Parcial
Custo	Depende da ferramenta	Depende da ferramenta	Depende da ferramenta

Comparação entre C, C++ e Java

Critérios gerais	C	C++	Java
Aplicabilidade	Sim	Sim	Parcial
Confiabilidade	Não	Não	Sim
Aprendizado	Não	Não	Não
Eficiência	Sim	Sim	Parcial
Portabilidade	Não	Não	Sim
Método de projeto	C e C++ são de domínio público. Java pertence à Oracle, porém é liberada gratuitamente. Existem inúmeras ferramentas gratuitas e pagas. Depende da escolha da equipe de desenvolvimento.		
Evolutibilidade			
Reusabilidade			
Integração			
Custo	Depende da ferramenta	Depende da ferramenta	Depende da ferramenta

Comparação entre C, C++ e Java

Crítérios específicos	C	C++	Java
Escopo	Sim	Sim	Sim
Expressões e comandos	Sim	Sim	Sim
Tipos primitivos e compostos	Sim	Sim	Sim
Gerenciamento de memória	Programador	Programador	Sistema
Persistência dos dados	Biblioteca de funções	Biblioteca de classes e funções	JDBC, biblioteca de classes, serialização
Passagem de parâmetros	Lista variável e por valor	Lista variável, default, por valor e por referência	Lista variável, por valor e por cópia de referência

Comparação entre C, C++ e Java

Crítérios específicos	C	C++	Java
Escopo	Sim	Sim	Sim
Expressões e comandos	As três LPs requerem a definição explícita de entidades, associando-as a um escopo de visibilidade. Há pequenas diferenças.		
Tipos primitivos e compostos			
Gerenciamento de memória	Programador	Programador	Sistema
Persistência dos dados	Biblioteca de funções	Biblioteca de classes e funções	JDBC, biblioteca de classes, serialização
Passagem de parâmetros	Lista variável e por valor	Lista variável, default, por valor e por referência	Lista variável, por valor e por cópia de referência

Comparação entre C, C++ e Java

Critérios específicos	C	C++	Java
Escopo	Sim	Sim	Sim
Expressões e comandos	Sim	Sim	Sim
Tipos primitivos e compostos	Todas oferecem uma ampla variedade de expressões e comandos.		
Gerenciamento de memória	Programador	Programador	Sistema
Persistência dos dados	Biblioteca de funções	Biblioteca de classes e funções	JDBC, biblioteca de classes, serialização
Passagem de parâmetros	Lista variável e por valor	Lista variável, default, por valor e por referência	Lista variável, por valor e por cópia de referência

Comparação entre C, C++ e Java

Critérios específicos	C	C++	Java
Escopo	Sim	Sim	Sim
Expressões e comandos	Sim	Sim	Sim
Tipos primitivos e compostos	Sim	Sim	Sim
Gerenciamento de memória	Todas oferecem ampla variedade de tipos primitivos (mas C não oferece booleano) e compostos (mas nenhuma oferece conjunto potência). Recentemente, Java 8 incluiu o tipo função na linguagem (vide <i>closures</i>).		
Persistência dos dados			
Passagem de parâmetros	Lista variável e por valor	Lista variável, default, por valor e por referência	Lista variável, por valor e por cópia de referência

Comparação entre C, C++ e Java

Crítérios específicos	C	C++	Java
Escopo	Sim	Sim	Sim
Expressões e comandos	Sim	Sim	Sim
Tipos primitivos e compostos	Sim	Sim	Sim
Gerenciamento de memória	Programador	Programador	Sistema
Persistência dos dados	C/C++ deixam a cargo do programador. Java utiliza coletor de lixo.		
Passagem de parâmetros	Lista variável e por valor	Lista variável, default, por valor e por referência	Lista variável, por valor e por cópia de referência

Comparação entre C, C++ e Java

Crítérios específicos	C	C++	Java
Escopo	Sim	Sim	Sim
Expressões e comandos	C/C++ oferecem funções de I/O, mas deixam persistência a cargo do programador. Não existe padrão para interface com BD. Java possui serialização e padronizou interface com BD no JDBC, além de ter operações de I/O.		
Tipos primitivos e compostos			
Gerenciamento de memória			
Persistência dos dados	Biblioteca de funções	Biblioteca de classes e funções	JDBC, biblioteca de classes, serialização
Passagem de parâmetros	Lista variável e por valor	Lista variável, default, por valor e por referência	Lista variável, por valor e por cópia de referência

Comparação entre C, C++ e Java

Critérios específicos	C	C++	Java
Escopo	Sim	Sim	Sim
Expressões e comandos	Sim	Sim	Sim
Tipos primitivos e compostos	Sim	Sim	Sim
Gerenciamento de memória	Programador	Programador	Sistema
Persistência dos dados	C usa apenas passagem por valor, obrigando o uso de ponteiros em diversas ocasiões. C++ oferece o maior leque de opções. Java incluiu <i>varargs</i> na versão 5.		
Passagem de parâmetros	Lista variável e por valor	Lista variável, default, por valor e por referência	Lista variável, por valor e por cópia de referência

Comparação entre C, C++ e Java

CrITÉrios especÍficos	C	C++	Java
Encapsulamento e proteÇ�o	Parcial	Sim	Sim
Sistema de tipos	N�o	Parcial	Sim
Verifica��o de tipos	Est�tica	Est�tica / Din�mica	Est�tica / Din�mica
Polimorfismo	Coer��o e sobrecarga	Todos	Todos
Exce��es	N�o	Parcial	Sim
Concorr�ncia	N�o (biblioteca de fun��es)	N�o (biblioteca de fun��es)	Sim

Comparação entre C, C++ e Java

Crítérios específicos	C	C++	Java
Encapsulamento e proteção	Parcial	Sim	Sim
Sistema de tipos	C oferece apenas encapsulamento de dados. Versões recentes permitem ocultamento com declaração (.h) & definição (.c). Java e C++ oferecem mecanismo de classes e pacotes.		
Verificação de tipos			
Polimorfismo	Coerção e sobrecarga	Todos	Todos
Exceções	Não	Parcial	Sim
Concorrência	Não (biblioteca de funções)	Não (biblioteca de funções)	Sim

Critérios específicos	C	C++	Java
Encapsulamento e proteção	Parcial	Sim	Sim
Sistema de tipos	Não	Parcial	Sim
Verificação de tipos	Em C, diversos mecanismos (ex.: uniões livres, coerções e aritmética de ponteiros) permitem violação do sistema de tipos. C++ herda isso, mas possui sistema de tipos mais rigoroso se usado em sua forma OO. Java possui um sistema de tipos bastante rigoroso.		
Polimorfismo			
Exceções			
Concorrência	Não (biblioteca de funções)	Não (biblioteca de funções)	Sim

Comparação entre C, C++ e Java

Crítérios específicos	C	C++	Java
Encapsulamento e proteção	Parcial	Sim	Sim
Sistema de tipos	Não	Parcial	Sim
Verificação de tipos	Estática	Estática / Dinâmica	Estática / Dinâmica
Polimorfismo	Todas as verificações de C são estáticas. C++ e Java fazem algumas verificações dinâmicas (ex.: amarração tardia, verificação de índice de vetor).		
Exceções			
Concorrência	Não (biblioteca de funções)	Não (biblioteca de funções)	Sim

Comparação entre C, C++ e Java

Crítérios específicos	C	C++	Java
Encapsulamento e proteção	Parcial	Sim	Sim
Sistema de tipos	Não	Parcial	Sim
Verificação de tipos	Estática	Estática / Dinâmica	Estática / Dinâmica
Polimorfismo	Coerção e sobrecarga	Todos	Todos
Exceções	C não possui polimorfismo paramétrico ou de inclusão. C++ e Java possuem todos, porém Java não permite sobrescrita de operadores.		
Concorrência			
	de funções)	de funções)	

Comparação entre C, C++ e Java

Crítérios específicos	C	C++	Java
Encapsulamento e proteção	Parcial	Sim	Sim
Sistema de tipos	Não	Parcial	Sim
Verificação de tipos	C não oferece. C++ oferece, mas não obriga seu uso. Java oferece um sistema bastante rigoroso de tratamento de exceções.		
Polimorfismo			
Exceções	Não	Parcial	Sim
Concorrência	Não (biblioteca de funções)	Não (biblioteca de funções)	Sim

Comparação entre C, C++ e Java

Critérios específicos	C	C++	Java
Encapsulamento e proteção	Parcial	Sim	Sim
Sistema de tipos	Não	Parcial	Sim
Verificação de tipos	Estática	Estática / Dinâmica	Estática / Dinâmica
Polimorfismo	Coerção e	Todos	Todos
Exceções	Java oferece recursos nativos para exclusão mútua (<i>synchronized</i>) e oferece <i>threads</i> em sua API básica.		
Concorrência	Não (biblioteca de funções)	Não (biblioteca de funções)	Sim

- Vários autores já fizeram comparações entre LPs;
- Procure saber:
 - Algumas referências no livro do Flávio;
 - Google Scholar, ACM DL, IEEE Xplorer, SpringerLink, etc.



[**http://nemo.inf.ufes.br/**](http://nemo.inf.ufes.br/)