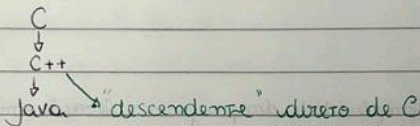


Programação Orientada a Objetos

Cap 1 "The history and evolution of Java"

- Capítulo explicativo referente a história de surgimento de Java.
- Java é uma linguagem de programação.
- As inovações da linguagem de programação ocorrem por 2 motivos:
 - ▷ Se adotar as mudanças e necessidades.
 - ▷ Implementar refinamentos e melhorias na publicação.

Linhaagem de Java:



- Muitos caracteres de Java vem de ambas as linguagens.
 - De C → sintaxe
 - De C++ → orientação a objeto
- O surgimento de Java está relacionado com as melhorias e os refinamentos de diversas linguagens.

• Nascimento da Programação Moderna: C

↳ A criação de C ocorreu devido uma necessidade de estrutura, eficiência e uma linguagem de alto desempenho que conseguisse substituir assembly na criação de programas.

▷ 3 compromissos para projetar uma linguagem de programação

1. Facilidade de usar vs poder;
2. Segurança vs eficiência;
3. Rigidez vs Extensibilidade.

- Antes de C os programadores tinham que escrever linguagens que produziam uma qualidade/inecuidade de auto
- As primeiras linguagens tinham um problema de estrutura, de forma os códigos eram impossíveis de entender (BASIC, COBOL e FORTRAN)
- Existem linguagens com estrutura mas com problema de eficiência (Pascal)
- Criação de C foi um marco, considerado o começo do uso moderno no programo
- ção
- Resultou no surgimento de C.

▷ C++

- ↳ Aumento da capacidade e complexidade dos programas levou à uma necessidade de uma linguagem melhor e mais complexa → C++
- Programação Orientada a Objetos: ajuda a organizar programas complexos por meio de herança, encapsulamento e polimorfismo
- C++ permitiu os programadores a entenderem e escreverem programas maiores
- C++ é uma extensão de C apartir do código de orientação de objetos

• Criação de Java

- ↳ objetivo original foi a necessidade de uma linguagem independente de plataforma destinada a dispositivos eletrônicos (micro-ondas e controles remotos)
- ↳ Java NÃO é uma versão para a internet de C++
- ↳ Influência por C++
- ↳ NÃO é uma versão aprimorada de C++
- ↳ C++ foi usado para resolver x problemas e Java y.
- ↳ Java trouxe certas mudanças, como: aprimoramento do paradigma orientado a objetos, suporte integrado a multithreading e uma biblioteca que facilitava o acesso à internet
- ↳ A linguagem como um todo tornou Java notável

• Conexão com C#

↳ C# influenciada por Java

↳ Semelhanças:

✓ Sintaxe semelhante

✓ Suporte à programação distribuída

✓ Uso de vários modelos de objetos

✓ Estrutura e estilo geral parecidos

• Como Java impactou a Internet

↳ simplificou a programação web

→ Introdução dos applets

↳ transformou a forma que o conteúdo online era visto

→ Resolveu 2 dos maiores problemas



• Portabilidade Segurança

• Java Applets

↳ permitiu que pequenos programas fossem transmitidos e executados diretamente em navegadores

↳ Maior parte do procedimento de verificação, tornando as páginas + interativas e dinâmicas

→ Os Applets foram um passo importante para mostrar o poder do Java

→ JDK 9 - começou a descontinuação dos applets

→ JDK 11 - removido

Segurança na Java

- Run em programas dinâmicos: cavalos de troia, vírus, códigos maliciosos
 - ↳ Podem acessar recursos do sistema de forma não autorizada
- Medidas de segurança Java: protegendo o usuário
 - ↳ Isola a aplicação em um ambiente de execução usando o acesso a partes variáveis do sistema
 - ↳ Executa somente funções permitidas
- Portabilidade + Segurança
 - ↳ funciona em diversos ambientes e SOs
 - ↳ Impacto Inovador:
 - ↳ + segurança
 - ↳ Desenvolvimento de aplicações seguras e dinâmicas

Portabilidade

- ↳ JVM (Java Virtual Machine) → mesmo código fonte executado em diferentes dispositivos
 - ↳ protege o sistema de códigos maliciosos

• Java se tornou ideal para aplicações web

Byte Code e Evolução Java

- Java Compiler gera byte code
 - ↳ conjunto de instruções otimizadas para a Java Virtual Machine
- JVM interpreta por si só o código
 - ↳ elimina necessidade de versões específicas para cada tipo de sistema

Segurança e Sandbox

- ↳ Ambiente seguro
 - ↳ ambiente onde o JVM executa o código

Desempenho e Just in time

- ↳ Hot Spot JIT Compiler converte partes do byte code para código nativo em tempo de execução
 - ↳ Não compromete a segurança

→ pequenos programas que rodam dentro de navegadores

• Fim dos Applets e Web Start

↳ Java plugin fica deixado de ser suportado pelos navegadores

→ JDK 11 descontinua os applets

→ Remoção do Java Web Start

→ HOJE: `yulink` (JDK 9) e `ypackage` (JDK 16) não usam para distribuir app. Java

• Cronograma de lançamento rápido → a cada 6 meses (março e setembro)

→ Feature releases com novas funcionalidades

→ Lançamento LTS (Long-Term Support)

• Vantagens do Novo cronograma

→ Aumento na disponibilidade de recursos

→ Resposta Rápida

• Servlets → + eficiente que CGI

→ pequenos programas executados no servidor para gerar conteúdo dinâmico

→ rodam em ambientes para servidores

→ SINTAXE de C e C++

Neutro em Aniquilando

"você uma vez, execute sempre"

Programas portáteis

Interpretado e alto desempenho

para byte code Java

compilação just in time

Distribuído

• PARA internet (TCP/IP)

• RMI

Dinâmico:

• informações importantes e rápido link dinâmico

→ Simple: fácil de aprender (experientes)

→ Object-Oriented: abordagem limpa e programática. Tudo é objeto e Não atropalha - equilíbrio

→ Robust: execução confiável e verificação rigorosa, gerenciamento automático de recursos e eliminação de erros comuns

Multi-threaded:

• execução simultânea de tarefas + sincronização e ambiente

Revolução do Java → Marcado por aprimoramentos e inovações

• Java:

→ 4.0 nasceu na programação

→ 4.1 novos elementos

→ 2 framework Swing e Collections Framework

• J2SE 4.3 e 4.4 melhor desempenho / 4.4 a palavra `assert`

→ 5 mudanças na linguagem

• JDK7 → simplificação de código e aumento de legibilidade e segurança

• Java tem sido um pilar de inovações desde sua criação, e evoluindo sempre