

Resumo da Evolução de Linguagens de Programação: De C a Java

A Origem e Evolução do Java

- A criação do Java foi impulsionada pela necessidade de resolver problemas que as linguagens anteriores não podiam, adaptando-se a ambientes em mudança e melhorando a programação.
- Ele herdou a sintaxe de C e recursos orientados a objetos de C++, misturando os melhores elementos de seus antecessores.
- O desenvolvimento do Java reflete esse legado de inovação e adaptação dentro da evolução da linguagem de computador.

O Impacto da Linguagem C

- A criação do C abordou de forma semelhante as limitações das linguagens anteriores, oferecendo uma alternativa poderosa, eficiente e amigável ao programador, o que promoveu a adoção generalizada.

O Surgimento e as Contribuições do C++

- C++, criado por Bjarne Stroustrup, abordou as limitações de C na gestão de programas complexos.
- Evolvendo de C e incorporando recursos de programação orientada a objetos, como herança, encapsulamento e polimorfismo, o C++ permitiu que os programadores lidassem com projetos maiores e mais intrincados do que antes.

Origens e Contexto

- Seu sucesso ~~foi~~ resultou da extensão das capacidades de C, em vez de substituí-lo completamente.
- No entanto, a ascensão da internet logo estimulou a evolução da ling. de programação Java.

O Nascimento do Java

- O Java, inicialmente chamado de Oak, foi criado em 1991 na Sun Microsystems para atender à necessidade de programação independente de plataforma em eletrônicos de consumo.
- A ascensão da World Wide Web aumentou significativamente a importância do Java, pois também exigia programas portáteis.

Diferencição e Sucesso do Java

- Embora influenciado por C e C++, o Java é distinto, resolvendo diferentes problemas e não pretendido como um substituto.
- Seu sucesso resultou de seu design holístico, abordando perfeitamente as demandas do emergente ambiente de computação distribuída.

Legado e Impacto do Java

- Java influenciou significativamente o C#, compartilhando sintaxe e modelos de objetos.
- O impacto do Java na internet inclui a criação de applets, programas dinâmicos executados em navegadores da Web, abordando preocupações de segurança e portabilidade.

Segurança e Portabilidade do Java

- Os recursos de segurança do Java impediram que programas baixados prejudicassem os sistemas, enquanto sua portabilidade permitiu que o mesmo código fosse executado em várias plataformas.
- A portabilidade e a segurança do Java decorrem do uso de bytecode, executado pelo Java Virtual Machine (JVM).
- O JVM permite que programas Java sejam executados em várias plataformas, ao contrário do código nativo.
- Embora a interpretação de bytecode seja mais lenta do que a execução de código nativo, a compilação Just-In-Time (JIT) otimizada do Java mitiga isso.

Obsolescência de tecnologias e novos ciclos de lançamento

- Applets e Java Web Start, outrora cruciais, estão obsoletos devido à falta de suporte no navegador e JDK, exigindo métodos alternativos de implantação como jlink e jpackage.
- O cronograma de lançamento do Java mudou para lançamentos de recursos semestrais e lançamentos de suporte de longo prazo (LTS) a cada três anos.

Atualizações e geração de conteúdo

- Isso permite atualizações de recursos mais rápidas e respostas oportunas às necessidades de programação em evolução.
- Servlets, rodando no servidor, permitem a geração dinâmica de conteúdo.

Objetivos e Recursos de Design

- Os principais objetivos de ~~essa~~ design do Java incluiriam simplicidade, segurança, portabilidade, design orientado a objetos, robustez, multithreading e alto desempenho.
- A robustez do Java decorre de recursos como gerenciamento automático de memória e tratamento de exceções.
- O sistema de tempo de execução do Java oferece sincronização de multiprocessos robusta.

Arquitetura e Portabilidade

- Sua arquitetura garante legibilidade e portabilidade do código.
- Java compila para bytecode, permitindo a execução multiplataforma com alto desempenho através de compilação just-in-time.
- Ele é projetado para ambientes distribuídos, suportando TCP/IP e Remote Method Invocation.

Início e Versões Iniciais

- A evolução do Java envolveu rápida inovação, começando com o lançamento revolucionário do Java 1.0 e continuando através de adições significativas e mudanças nas versões subsequentes.

J2SE 5 e Adições Subsequentes

- Grandes atualizações como o J2SE 5 introduziram recursos como genéricos e anotações, alterando fundamentalmente a linguagem.
- Versões posteriores construíram sobre essa base, adicionando melhorias incrementais e novas APIs, mantendo a compatibilidade com versões anteriores.

Melhorias no Java SE 7

- O Java SE 7 melhorou significativamente o framework NIO (NIO.2) e introduziu o Fork/Join Framework.
- Este último simplifica a programação paralela, permitindo a execução de tarefas simultâneas e a utilização automática de multiprocessadores para um desempenho aprimorado.

Inovações no Java SE 8

- Ele introduziu expressões lambda, impactando a estrutura do código e adicionando recursos de programação funcional.

Lançamentos e Atualizações do JDK

- Lançamentos do Java SE 9 (com módulos e JShell), Java SE 10 (inferência local de tipo de variável) e Java SE 11 (HTTP Client API e remoção de applets), esses focaram em aprimoramentos e atualizações de API.
- O JDK 14 adicionou suporte a expressão switch.
- Applets e Java Web Start foram removidos.

Recursos adicionados a partir de JDK 15

- A evolução do Java, particularmente desde o JDK 15, inclui recursos como blocos de texto, instâncias aprimoradas, registros, pacote e classes seladas (JDK 17).

Inovação e Influências no Desenvolvimento do Java

- O código aberto alimentou ainda mais o desenvolvimento e a inovação contínuas do Java.
- A história do Java é de constante inovação, começando com seu impacto na programação da internet e a introdução da JVM e do bytecode para segurança e portabilidade.
- O JCP moldou ainda mais sua evolução, resultando no ~~último~~ último lançamento do JDK 17.