Boa! 💡 Você quer desafios **hardcore** que forcem bastante o raciocínio lógico e a aplicação prática desses conceitos básicos de programação. Vou propor **5 desafios progressivos** (nível difícil mesmo), cada um atacando de um jeito diferente os temas:

## **🔥 1. Tipos Primitivos – “Máquina de Precisão”**

Crie um programa que:

* Leia **dois números decimais muito grandes** (ex: double ou BigDecimal se disponível).
* Converta-os em **float** e mostre a diferença entre os valores originais e os convertidos.
* Implemente uma função que calcule o **erro percentual de precisão** entre os tipos.
* O desafio é perceber como cada tipo primitivo perde dados dependendo do tamanho.

🎯 **Extra:** Faça testes com números enormes e muito pequenos (ex: 0.00000000123).

## **🔥 2. Convenções de Variáveis – “Nomeador de Código Espaguete”**

* Receba um código-fonte simples (como uma String com várias linhas).
* Identifique variáveis mal nomeadas (ex: a, x1, data2) e sugira nomes mais adequados seguindo convenções (camelCase, snake\_case, etc.).
* Gere uma “tabela de renomeação” mostrando nome\_antigo -> nome\_sugerido.

🎯 **Extra:** implemente regras contextuais, por exemplo: se encontrar idade = 20;, sugerir idadePessoa.

## **🔥 3. Declaração e Tamanho em Memória – “Medidor de Peso”**

* Implemente um programa que leia um **array com milhões de elementos** de tipos diferentes (int, long, float, double, boolean, char).
* Calcule **quanto de memória total está sendo usado** considerando o tamanho de cada tipo.
* Exiba um relatório no formato:

Tipo: int | Tamanho: 4 bytes | Quantidade: 1\_000\_000 | Total: 4 MB

...

* Compare a memória usada se o mesmo array fosse de outro tipo.

🎯 **Extra:** Calcule também o tempo de execução para operações matemáticas simples em cada tipo.

## **🔥 4. Casting – “Conversor de Caos”**

* Leia uma entrada numérica (pode ser double grande).
* Converta em todos os tipos possíveis (byte, short, int, long, float).
* Mostre os resultados e explique os **estouros (overflow)** ou **perda de dados**.
* Exemplo: double 130.75 → byte = -126.

🎯 **Extra:** implemente um “simulador de estouro”: dado um número, mostre a representação binária antes e depois do casting.

## **🔥 5. Strings – “Analisador de Palíndromos Hardcore”**

* Receba um texto longo (ex: um parágrafo).
* Verifique se **cada palavra é um palíndromo parcial** (ignorar acentos, maiúsculas, pontuação).
* Crie uma função que calcule a **similaridade palindrômica**:
  + 100% → palavra igual ao inverso.
  + 0% → nenhuma letra coincide.
  + Exemplo:
    - "ovo" → 100%
    - "casa" → 25% (apenas ‘a’ coincide no meio).
* Mostre a pontuação geral do texto.

🎯 **Extra:** otimizar para funcionar com textos grandes sem explodir em memória.

👉 Esses 5 desafios vão te forçar a pensar em: **precisão, boas práticas, performance, overflow e manipulação avançada de Strings**.

Quer que eu já monte **esqueleto de código em Java** para cada desafio (com as assinaturas das funções) para você só implementar a lógica?