Návrh a implementace Rate Limiteru

- **Scénář:** Představte si, že vyvíjíte vysoce dostupnou REST API službu. Abychom předešli zneužití a zajistili stabilitu systému, potřebujeme implementovat mechanismus pro omezování počtu požadavků (Rate Limiting).
- **Cíl:** Navrhněte a popište implementaci "Rate Limiter" služby v Javě.

Část 1: Základní design (Single-node)

Začněme se zjednodušenou verzí, kde celá naše aplikace běží na jediném serveru (jedné instanci JVM).

1. **Definice API:** Jak byste navrhl(a) rozhraní (interface) pro tuto službu? Jaká by byla signatura hlavní metody?
2. **Volba algoritmu:** Jaký algoritmus byste použil(a) pro sledování požadavků? Krátce zdůvodněte svou volbu.
3. **Datová struktura:** Jakou datovou strukturu byste v Javě použil(a) pro ukládání stavu (počtu požadavků) pro jednotlivé klienty?
4. **Concurrency:** Naše služba bude zpracovávat mnoho požadavků současně z různých vláken. Jak zajistíte, že vaše implementace je thread-safe (bezpečná pro práci ve více vláknech) a zároveň výkonná?

Část 2: Distribuovaný systém (Multi-node)

V reálném světě naše služba poběží ve více instancích (např. v Kubernetes clusteru) pro zajištění vysoké dostupnosti a škálovatelnosti.

1. **Problém:** Jaký problém nastane s implementací z Části 1, pokud ji nasadíme na více serverů za load balancer?
2. **Řešení:** Jak byste upravil(a) svůj návrh, aby Rate Limiting fungoval konzistentně napříč všemi instancemi?

Část 3: Integrace a produkční nasazení

1. **Konfigurace:** Jak byste navrhl(a) systém tak, aby bylo možné snadno měnit limity (např. "100 požadavků za minutu pro klienta A", "20 za sekundu pro klienta B")? **Bonus:** Jak to udělat bez nutnosti restartu aplikace?
2. **Testování:** Jak byste otestoval(a) funkčnost této komponenty? Zejména jak byste testoval(a) její chování pod vysokou zátěží a ve vícevláknovém prostředí?

Vyřešení cyklických závislosti

- **Zadání:** Jak bystě vyřešil když třída A závisí na třídě B a třída B zároveň závisí na třídě A (případně přes více kroků: A -> B -> C -> A)?
- **Bonus:** jak to lze vyřešit ve Springu?

Indirect Cyclic Dependency