

## Zadanie 2: Kalkulačka

Úlohou je implementovať textovú kalkulačku v jazyku Java. Na vstupe je reťazec znakov reprezentujúci výraz, na výstupe má byť vypočítaná hodnota výrazu.

Podporované majú byť:

- celé čísla,
- operátory +, -, \*, / (celočíselné delenie),
- unárne mínus,
- zátvorky s ľubovoľnou úrovňou vnorenia.

Je nutné manuálne vytvoriť syntaktický analyzátor, teda nie je možné využiť tzv. generátory parserov. Návrh analyzátoru musí zodpovedať bezkontextovej gramatike dodanej k projektu. Konkrétnu gramatiku predstaví pedagóg na cvičení. Použitie má byť syntaxou riadené vyhodnocovanie (interpretovanie) hodnoty výrazu.

Do repozitára, ktorý už máte vytvorený (z 1. zadania), pomocou Gitu nahrajte svoj kód najneskôr do začiatku vášho 9. cvičenia. Pridaný obsah má mať nasledujúcu štruktúru:

```
zadanie2/  
  src/  
  README.txt  
  pom.xml alebo build.gradle
```

Nezabudnite predovšetkým na vytvorenie priečinku `zadanie2`. Súbor `README.txt` má obsahovať gramatiku v tvare EBNF. Projekt by mal obsahovať konfiguráciu tzv. build nástroja: buď `pom.xml` (pre Maven) alebo `build.gradle`, resp. `build.gradle.kts` (Gradle), prípadne `build.xml` (Ant). Adresárová štruktúra samotných zdrojových súborov nie je presne daná, mala by zodpovedať konvenciám vybraného kompilačného nástroja.

K zadaniu vytvorte aj jednotkové testy, napr. pomocou knižnice [JUnit](#) alebo inej. Očakáva sa spolu rádovo 10 volaní funkcií "assert". Viac ako o počet však ide o presnosť - snažte sa otestovať hlavne hraničné situácie, teda napr. chybné reťazce, ktoré sa od správnych líšia len jedným znakom, vstupy s množstvom vnorených zátvoriek, reťazec s medzerami, vstup kde záleží na priorite operátorov a pod. Pripravte si prostredie tak, aby ste na požiadanie cvičiaceho spustili testy, napr. príkazom `mvn test` alebo pomocou IDE. Zadanie je možné odovzdať aj bez testov, no za znížený počet bodov.