

Logging - zadania

Spis treści

Zadanie 1	1
Zadanie 2	1
Zadanie 3*	1

Zadanie 1

Jak będzie wyglądała klasa poniżej bez wykorzystania Lomboka?

```
@Slf4j
public class SLF4JLogging {
    // reszta kodu
}
```

Zadanie 2

Napisz konfigurację Logback, w której określisz, że:

- W konsoli chcesz logować wszystkie zdarzenia i logujesz je z poziomem INFO
- Do tego chcesz skonfigurować logowanie do plików. Pliki mają się rollować się przy rozmiarze 20MB. Jeżeli pliki z logami przekroczą rozmiar 10GB, mamy zacząć usuwać najstarsze z nich. Pliki z logami mają być przetrzymywane przez 50 dni. Logowanie do plików ma mieć poziom INFO. Logi w plikach mają być zapisywane w katalogu `my-logs`. Pliki archiwalne mają być umiejscowione w katalogu `my-logs/archived`. Najświeższy plik z logami ma mieć nazwę `newest.log`. Pliki archiwalne mają mieć nazwę, np. `rolled.{data}.{numer_porządkowy}.log`. Do plików mamy logować zdarzenia z całej aplikacji, czyli z jej głównej paczki, np. `pl.zajavka`.

Zadanie 3*

**Gwiazdka w tym poleceniu oznacza, że rozwiązanie tego polecenie będzie Ci zaprezentowane tylko w formie pisemnej. Programista/-ka czyta bardzo dużo gotowego kodu, dlatego czasami nie będziemy nagrywać rozwiązań zadań, żeby dać Ci opcję pracy z gotowym kodem.*

Przygotuj mini projekt w celu poćwiczenia logowania. Stwórz klasę `BankAccount` reprezentującą konto bankowe. Konto powinno zawierać pola:

- `id` - unikalny identyfikator konta,
- `owner` - imię i nazwisko właściciela konta,
- `balance` - saldo konta.

Stwórz klasę `BankTransaction` reprezentującą pojedynczą transakcję na koncie bankowym. Transakcja powinna zawierać pola:

- `id` - unikalny identyfikator transakcji,
- `account` - konto, na którym odbyła się transakcja,
- `amount` - kwota transakcji,
- `date` - data transakcji.

Stwórz klasę `BankService` reprezentującą serwis bankowy. Serwis powinien zawierać metody do otwierania nowych kont, wyszukiwania kont, wykonywania operacji wpłaty, wypłaty i przelewu oraz do pobierania historii transakcji dla danego konta.

W klasie `BankService` wykorzystaj framework Logback i abstrakcję `Slf4j` do logowania informacji o działaniu serwisu bankowego. Wprowadź logowanie informacji o otwieraniu nowych kont, wpłatach, wypłatach, przelewach i pobieraniu historii transakcji.

Po stworzeniu powyższych klas i napisaniu kodu realizującego powyższe polecenie zapoznaj się z poniższym przykładowym logiem:

```
2022-08-01 12:00:00,001 [main] INFO BankService - Making transfer 1000.0 from account with id 14124 to account with id 9123221...
2022-08-01 12:00:00,002 [main] INFO BankService - Transfer of 1000.0 made from account with id 14124 to account with id 9123221.
2022-08-01 12:00:00,002 [main] DEBUG BankService - Registered transaction 55231235 for accounts with ids 14124 i 9123221.
2022-08-01 12:00:00,133 [main] INFO BankService - Making transfer 1000.0 from account with id 3984755 to account with id 887234...
2022-08-01 12:00:00,134 [main] ERROR BankService - Account 887234 has an insufficient balance.
2022-08-01 12:00:01,000 [main] INFO BankService - Making transfer 1000.0 from account with id 662344 to account with id 9128485...
2022-08-01 12:00:01,001 [main] INFO BankService - Transfer of 1000.0 made from account with id 662344 to account with id 9128485.
2022-08-01 12:00:01,003 [main] DEBUG BankService - Registered transaction 2378438 for accounts with ids 662344 i 9128485.
```

W tym logu został zaszyty błąd. Na czym polega ten błąd? Oczywiście odpowiedź znajdziesz w pliku z odpowiedziami.