

1. Projekt oks-01

Základní informace:

- **Účel:** využití **logování z Log4j2**, příprava vlastní externí konfigurace
- **Kostr:** oks-data-01.zip
- **Odevzdávaný soubor:** oks-01.jar

Zadání:

- do připravené již plně fungující aplikace doplňte logovací příkazy
- připravte XML soubor externí konfigurace pro Log4j2

Popis vstupních dat:

- stáhněte si soubor oks-data-01.zip a rozbalte jej
 - všechny soubory jsou v kódování UTF-8 bez BOM
- v adresáři data se nacházejí vstupní textové soubory, které aplikace zpracovává
 - adresář obsahuje následující soubory:
 - ♦ priklady-oks-01-1.txt - vstupní soubor se správnými i chybnými daty
 - ♦ priklady-oks-01-2.txt - vstupní soubor pouze se správnými daty
- v adresáři lib se nachází knihovny Log4j
 - adresář obsahuje následující soubory:
 - ♦ log4j-api-2.11.1.jar - API knihovny
 - ♦ log4j-core-2.11.1.jar - jádro knihovny
 - oba soubory budete importovat do IntelliJ
- v adresáři oks01 se nachází kompletní funkční aplikace pro generování osobních čísel na ZČU
 - adresář obsahuje následující podadresáře a soubory:
 - ♦ doc - kompletní JavaDoc dokumentace
 - ♦ src/oks01 - zdrojové soubory aplikace
 - ♦ oks-01-pgm.jar - spustitelný soubor aplikace
 - funkčnost aplikace si můžete ověřit zadáním příkazu:

```
java -jar oks-01-pgm.jar ../data/priklady-oks-01-1.txt fav
```

 - ♦ vznikne soubor vysledky.txt

- dále se zde nachází soubor `oks-01-konfigurace.xml` - přejmenovaný soubor defaultní konfigurace Log4J
 - ♦ ten budete dle dalších pokynů upravovat
- v adresáři `kontrola` se nachází pomocný soubor pro kontrolní spuštění
 - adresář obsahuje následující soubory:
 - ♦ `kontrola.bat` - soubor s příkazy pro opakované spuštění aplikace
 - můžete využít pro kontrolu, pokud používáte Windows
- v adresáři `kontrola/vysledek` se nachází výsledky logování, které musíte dosáhnout
 - adresář obsahuje následující soubory:
 - ♦ `vzor-oks-01-log-obrazovka-all.txt` - logovací soubor celé aplikace v úrovni `Level.ALL` při spuštění se vstupním souborem `priklady-oks-01-1.txt`
 - ♦ `vzor-oks-01-log-info.txt` - logovací soubor aplikace v úrovni `Level.INFO` při postupném spuštění aplikace se vstupními soubory `data/neexistujici.txt`, `data/priklady-oks-01-2.txt` a `data/priklady-oks-01-1.txt`
 - ♦ `vzor-oks-01-log-debug.txt` - logovací soubor aplikace v úrovni `Level.DEBUG` při spuštění se vstupním souborem `priklady-oks-01-1.txt`
 - ♦ `vzor-oks-01-log-trace.txt` - logovací soubor aplikace v úrovni `Level.TRACE` při spuštění se vstupním souborem `priklady-oks-01-1.txt`

Note

Tyto soubory (kromě `vzor-oks-01-log-obrazovka-all.txt`) jsou rozhodující - vámi upravená aplikace musí vytvořit zcela identické logovací soubory (samozřejmě bez počátečního `vzor-`) - ty budou kontrolovány validátorem.

Warning

Pozor na skutečnost, že zde uvedené soubory logů jsou v kódování UTF-8 bez BOM. Pokud je budete vytvářet spuštěním přímo z IntelliJ, pak je dostanete také v UTF-8. Vytvoříte-li je však spuštěním z příkazové řádky, budou v nativním kódování platformy, tj. např. windows-1250. Pak je nutné je před porovnáním převést do UTF-8 například otevřením v Notepadu a uložením v kódování UTF-8.

Postup řešení:

- v IntelliJ založte nový projekt `oks-prj-01`
 - nastavte celému projektu kódování UTF-8
- File / Settings / Editor / File encodings / Project encoding: UTF-8**
- do adresáře `src` projektu přetáhněte celý adresář `oks01`
- přetáhněte soubor `data/priklady-oks-01-1.txt` do adresáře projektu `oks-prj-01`

- otevřete třídu Hlavni a spusťte ji

- ◆ na konzoli by se mělo objevit

```
Nezadáno jméno vstupního souboru nebo fakulta
Generátor osobních čísel
Použití:
java Hlavni jmeno_vstupniho_souboru.txt fakulta
```

- v **Run / Edit Configuration / ProgramArguments** zadejte `priklady-oks-01-1.txt` fav
- po opětovném spuštění programu se v `oks-prj-01` objeví soubor `vysledky.txt`

- přetáhněte soubor `oks01/oks-01-konfigurace.xml` do adresáře projektu `oks-prj-01`

- do Hlavni přidejte

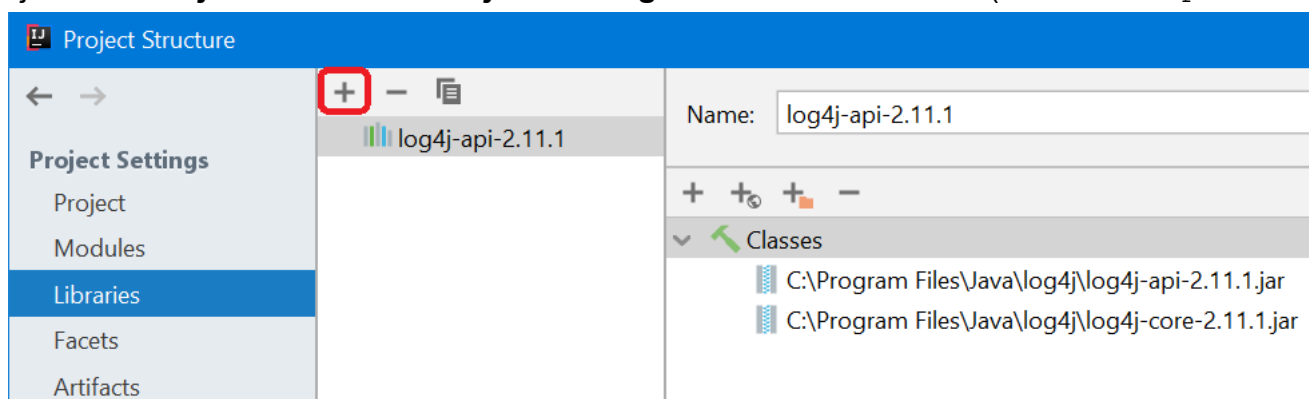
```
public class Hlavni {
    static {
        System.setProperty("log4j.configurationFile",
                           "oks-01-konfigurace.xml");
    }
    public static final Logger hlavniLogger = LogManager.getLogger();
}
```

- do `Hlavni.main()` přidejte

```
public static void main(String[] args) {
    hlavniLogger.info("Zacatek programu");
    parametryPrikazoveRadky(args);
}
```

- z adresáře `lib` přidejte soubory `log4j-api-2.11.1.jar` a `log4j-core-2.11.1.jar` do *Classpath*

použijte **File / Project Structure / Project Settings / Libraries / + / Java** (viz `oks-04.pdf` str.6)



- umístěte kurzor na červeně vysvícený Logger a pomocí **Alt + Enter** doplňte importy

```
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
```

- po puštění aplikace byste měli vidět na konzoli

```
17:14:32.806 [main] INFO oks01.Hlavni - Zacatek programu
```

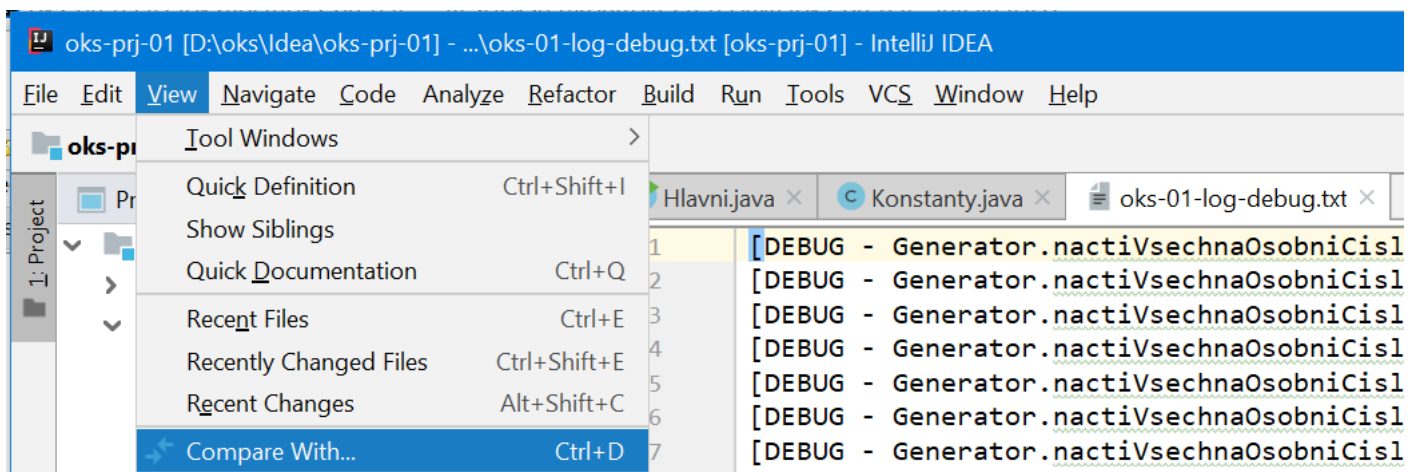
- pokud tento výpis nevidíte, udělali jste v postupu chybu
- editujte soubor `oks-01-konfigurace.xml` a opravujte jeho formát výpisu tak dlouho, dokud se nebude shodovat s požadovaným formátem, který zjistíte ze souboru `oks-01-log-obrazovka-all.txt`
 - výpis začíná úrovní
 - pak je jméno loggeru (nebo jméno třídy) zkrácené, tj. bez balíku
 - následuje jméno metody
 - ♦ pokud vás zarazí `OsobniCislo.<init>()`, pak vězte, že je to název konstruktoru (to byste měli vědět z KIV/OOP)
 - dále je vypisována logovací zpráva
 - řetězec končí platformově závislým odřádkováním
- po spuštění aplikace byste měli vidět na konzoli

```
[INFO - Hlavni.main()] - Zacatek programu
```

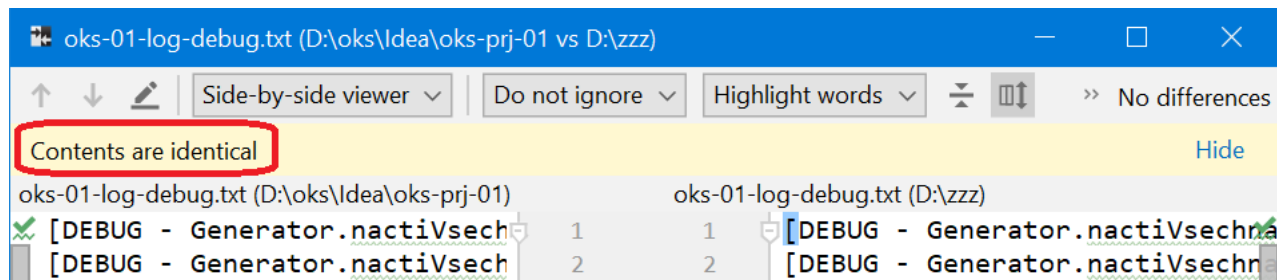
- z logovacího souboru `vzor-oks-01-log-obrazovka-all.txt` tedy vidíte, do jaké metody máte umístit logger konkrétní úrovně a s jakou konkrétní zprávou
 - doplňování logů je tak víceméně mechanická práce
 - postupujte tak, že nejdříve doplníte (a zkontrolujete) všechny logy do třídy `Hlavni`
 - pak ve třídě `Generator` definujete statický logger (podobně jako v `Hlavni`)
 - do třídy `Generator` doplňte všechny logy
 - pak pokračujte stejně se třídou `OsobniCislo`, tj. definujte logger a následně doplňte logy
- díky nastavení v `oks-01-konfigurace.xml` jsou všechny logy všech loggerů na obrazovku
 - předchozí bod opakujte tak dlouho, dokud nedostanete na obrazovce výpis totožný s obsahem souboru `vzor-oks-01-log-obrazovka-all.txt`

Note

Pro porovnání obsahu souborů můžete využít v IntelliJ službu **Viev / Compare With**



kdy musíte dostat



- veškerou další činnost budete provádět editováním souboru `oks-01-konfigurace.xml`
 - postupujte iterativně, nesnažte se napsat celou konfiguraci najednou
- formát uložte do `<Property>` (budete jej potřebovat nejméně třikrát)
- najednou se bude logovat do tří souborů - jejich jména (bez `vzor-`) budou opět v `<Property>`
 - `oks-01-log-info.txt` - sem se zapisuje v režimu *append* a jen do úrovně `Level.INFO`

prakticky to znamená, že sem zapisují současně loggery z `Hlavni` a z `Generator`

protože však `Generator` použít až do úrovně `DEBUG` (viz dále), je třeba tuto úroveň odfiltrovat pomocí `<ThresholdFilter>`

 - `oks-01-log-debug.txt` - soubor se při každém běhu přemazává (je jen pro ladění) a přijímá jen do úrovně `DEBUG` - tj. současně loggery z `Generator` a z `OsobniCislo`
 - `oks-01-log-trace.txt` - soubor se při každém běhu přemazává (je jen pro ladění) a přijímá až do úrovně `TRACE`, ovšem pouze z `OsobniCislo`
- možná budete mít problémy s dvojitým výpisem stejného logu - pak se jedná o aditivitu
- v `<Root>` zakomentujte připojení `Console`, čímž zrušíte výpisy na konzoli
 - to znamená, že při spuštění se budou vytvářet jen logovací soubory a soubor výsledků `vysledky.txt`

Kontrola úplnosti řešení:

- adresář `src` z IntelliJ překopírujte někam do svého pomocného adresáře, např. `D:\zzz`
 - dále sem překopírujte obsah adresářů `data` a `lib`

- nakonec sem z IntelliJ překopírujte váš soubor `oks-01-konfigurace.xml`

■ v tomto adresáři použijte příkaz:

```
javac -d . -cp log4j-api-2.11.1.jar;log4j-core-2.11.1.jar;. -encoding UTF-8 ►  
src/oks01/*.java
```

- Pozor - pokud pracujete v Linuxu, bude zřejmě nutné v nastavení *classpath* změnit oddělovač souborů `;`;

```
-cp log4j-api-2.11.1.jar;log4j-core-2.11.1.jar;.
```

na :

```
-cp log4j-api-2.11.1.jar:log4j-core-2.11.1.jar:.
```

- překlad je potřebný pro ověření správné funkčnosti dále zmíněným dávkovým souborem `kontrola.bat`
- program třikrát spusťte nebo použijte jednorázově `kontrola.bat`

Warning

Při spuštění nezapomeňte na to, že vámi vytvořený konfigurační soubor musí být ve stejném adresáři jako spouštěný `kontrola.bat` soubor.

- ♦ pokud nebudou vzniklé logovací soubory identické (po případném překódování do UTF-8 - viz výše) s příslušnými soubory adresáře `kontrola/vysledek`, tak ani neodevzdávejte a hledejte příčinu rozdílu

Příprava souborů k odevzdání:

■ v adresáři, ve kterém byla prováděna kontrola, použijte příkaz:

```
jar cMf oks-01.jar src oks-01-konfigurace.xml
```

■ výsledkem bude soubor `oks-01.jar`, který budete odevzdávat

- opravdu není spustitelný, neobsahuje `.class` soubory ani Javadoc
- jeho velikost by měla být do 10 KB - pokud je větší, je v něm něco navíc

■ po úspěšném odevzdání na Portále, proklikněte výsledné OK a, prosím, vyplňte v tabulce časovou náročnost této úlohy