

1. Projekt oks-03

Základní informace:

- **Účel:** prakticky se seznámit s pokročilými možnostmi jednotkových testů a logování jejich výsledků
- **Kostrá:** oks-data-03.zip
- **Odevzdávaný soubor:** oks-03.jar

Zadání:

- připravte parametrizované JUnit testy a logujte jejich výsledky, tj. využijte pokročilé možnosti jednotkových testů

Popis vstupních dat:

- stáhněte si soubor oks-data-03.zip a rozbalte jej
 - všechny soubory jsou v kódování UTF-8 bez BOM
- v adresáři src/oks03 se nachází zdrojové .java soubory
 - tento adresář budete celý importovat do IntelliJ a nebudete jej nijak měnit
 - adresář obsahuje následující soubory:
 - ♦ OsobniCislo.java - třída, kterou jste logovali a nad kterou jste psali JUnit testy - tento soubor je téměř totožný se souborem z projektu oks-01
 - liší se pouze jménem balíku
 - ♦ Konstanty.java - třída konstant
 - ♦ TypStudia.java - výčtový typ
- v adresáři test/oks03 se nachází zdrojový .java soubor
 - tento adresář budete celý importovat do IntelliJ a budete jej doplňovat o další soubory
 - adresář obsahuje následující soubor:
 - ♦ Kontrola_Prj_03.java - třída pomocí níž budete spouštět testy
 - tento soubor nebudete nijak měnit
- v adresáři data se nachází vstupní datový soubor
 - adresář obsahuje následující soubor:
 - ♦ priklady-oks-03.txt - data jsou velmi podobná těm z projektu oks-01
 - každý řádek je rozšířen o jeden údaj, který představuje očekávaný výsledek testu metody OsobniCislo.isPlatnyFormat()
 - pro účely ověření správnosti je na několika místech tento údaj zadán chybně

– tento soubor budete pouze číst a nebudete jej nijak měnit

■ v adresáři `kontrola` se nachází kontrolní výsledky

- adresář obsahuje následující podadresáře a soubor:
 - ♦ `oks03` - adresář s testovanými `.class` soubory
 - ♦ `vysledek` - adresář se vzorovým souborem výsledku `vzor-oks-03-log.txt`
 - ♦ `kontrola.bat` - překlad a spuštění kontrolního programu

■ v adresáři `lib` se nachází potřebné knihovny pro JUnit 5 a Log4j2

- adresář obsahuje deset `.jar` souborů

Postup řešení:

■ v IntelliJ založte nový projekt `oks-prj-03`

- nastavte celému projektu kódování UTF-8

File / Settings / Editor / File encodings / Project encoding: UTF-8

- do adresáře `src` projektu přetáhněte celý adresář `src/oks03`
 - ♦ tento adresář ani soubory v něm nebudete nijak měnit ani doplňovat
- vytvořte adresář `test` a pomocí kontextového menu a **Mark Directory as / Test Sources Root** jej nastavte jako adresář testů
 - ♦ přetáhněte do něj celý adresář `test/oks03`
 - ♦ většinu aktivit budete provádět v tomto adresáři
- přetáhněte soubor `data/priklady-oks-03.txt` do adresáře projektu `oks-prj-03`

■ prohlédněte si třídu `Kontrola_Prj_03.java`

- tu nebudete nijak měnit, ale je v ní mj. uvedeno pět důležitých názvů:
 - ♦ `oks-03-konfigurace.xml` - jméno konfiguračního souboru pro Log4j2
 - ♦ `logger` - jméno referenční proměnné instance loggeru
 - ♦ `datovy.zdroj.oks03` - klíč udávající jméno datového souboru
 - ♦ `oks03.OsobniCisloTest.class` - jméno testovacího souboru
 - ♦ `LogovaniVysledku` - jméno třídy, pomocí níž budete logovat výsledky

■ přidejte všechny `.jar` soubory z adresáře `lib` do *Classpath*

- postup je popsán v projektu `oks-01`

■ vytvořte třídu `OsobniCisloTest` s jednou metodou testující metodu `isPlatnyFormat()` se signaturou

```
@ParameterizedTest(name = "{index}: vysledek={0}; data:{1}")
@MethodSource("oks03.PripravaDatovehoZdroje#listDvojiceBooleanString")
void isPlatnyFormat(Boolean vysledek, String radkaDat) {
```

- skutečné parametry této metody budou dodávány z datového zdroje jako dvojice objektů
- metoda vytvoří instanci osobního čísla pomocí parametru `radkaDat`
- metoda pak pomocí parametru `vysledek` porovná, zda je platný formát

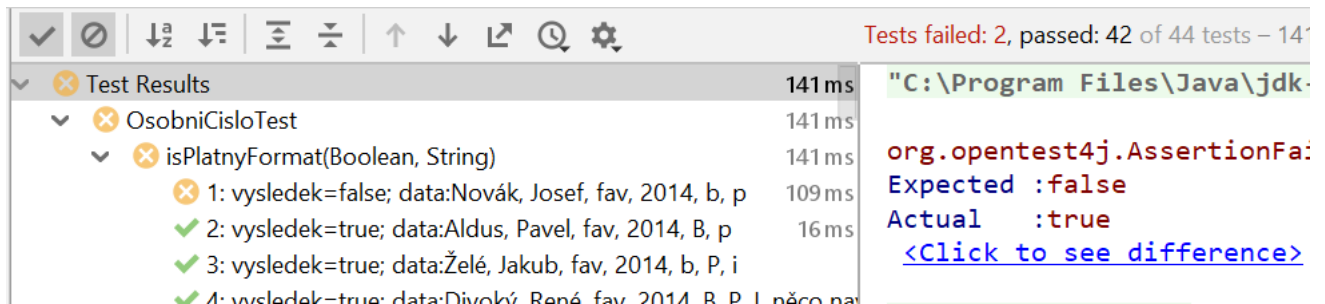
■ vytvořte třídu `PripravaDatovehoZdroje` se dvěma metodami:

- `private static List<String> nacteniSouboru()`
 - ♦ pomocí `String soubor = System.getProperty("datovy.zdroj.oks03");` zjistí jméno datového souboru
 - tuto řádku prozatím zakomentujte a pro účely ladění použijte provizorní:

```
String soubor = "prikklady-oks-03.txt";
```
 - ♦ načte tento soubor do seznamu řádek, ve kterém ponechá jen platné řádky
 - tj. vynechá prázdné řádky a řádky komentářů začínající `#`
 - ♦ tento seznam metoda vrátí
- `static List<Object[]> listDvojiceBooleanString()`
 - ♦ pomocí metody `nacteniSouboru()` získá seznam platných datových řádek
 - ♦ vytvoří `List<Object[]> generovano`, do kterého pro každou řádku uloží dvojici objektů typu `Boolean` a `String`

■ spusťte test `OsobniCisloTest.isPlatnyFormat(Boolean vysledek, String radkaDat)`

- výsledek by měl být



Tests failed: 2, passed: 42 of 44 tests – 141 ms

| Test Case | Duration | Status | Message |
|--|----------|--------|---|
| OsobniCisloTest | 141 ms | Failed | |
| isPlatnyFormat(Boolean, String) | 141 ms | Failed | |
| 1: vysledek=false; data:Novák, Josef, fav, 2014, b, p | 109 ms | Failed | org.opentest4j.AssertionFailedError: Expected :false Actual :true |
| 2: vysledek=true; data:Aldus, Pavel, fav, 2014, B, p | 16 ms | Passed | |
| 3: vysledek=true; data:Želé, Jakub, fav, 2014, b, P, i | | Passed | |
| 4: vysledek=true; data:Divoký, René, fav, 2014, B, P, i, něco na | | Passed | |

■ po úspěšném testu odkomentujte v metodě `nacteniSouboru()` řádku

```
String soubor = System.getProperty("datovy.zdroj.oks03");
```

- pokud to neuděláte, lokálně bude vše fungovat, ale validátor ohlásí chybu

■ připravte třídu `LogovaniVysledku`, která implementuje rozhraní `TestExecutionListener`

- ve třídě stačí překrýt metody `executionStarted()` a `executionFinished()`

- u obou metod nejprve vypíšte výsledky testu do konzole pomocí `System.err`
- po spuštění `main()` ze třídy `Kontrola_Prj_03` by se mělo v konzoli zobrazit zhruba (přesný formát není důležitý):

```

Hlavni_SpusteniTestu x
"C:\Program Files\Java\jdk-11.0.1\bin\java.exe" ...
1: vysledek=false; data:Novák, Josef, fav, 2014, b, p
  FAILED: expected: <false> but was: <true>
2: vysledek=true; data:Aldus, Pavel, fav, 2014, B, p
3: vysledek=true; data:Želé, Jakub, fav, 2014, b, P, i
4: vysledek=true; data:Divoký, René, fav, 2014, B, P, I, něco navíc
5: vysledek=true; data:Platnost, Karel, fav, 2014, n, p
6: vysledek=true; data:Plát, Josef, Fav, 2014, n, p

```

- do třídy `LogovaniVysledku` doplňte logování pomocí loggeru, jehož instance byla vytvořena ve třídě `Kontrola_Prj_03`
 - začátek testu bude v levelu `INFO`
 - v případě selhání testu bude informace v levelu `ERROR`
- v adresáři projektu `oks-prj-03` založte soubor `oks-03-konfigurace.xml`
 - bude obsahovat velmi jednoduchou konfiguraci pro Log4j2 pro jeden logger
 - vytvářený logovací soubor se bude jmenovat `oks-03-log.txt`
 - ♦ při každém spuštění se bude přemazávat
 - ♦ jeho očekávaný obsah je vidět v souboru `kontrola/vzor-oks-03-log.txt`
- po spuštění `main()` ze třídy `Kontrola_Prj_03` by měl být obsah souboru `oks-03-log.txt`

```

[INFO] - 1: vysledek=false; data:Novák, Josef, fav, 2014, b, p
[ERROR] - expected: <false> but was: <true>
[INFO] - 2: vysledek=true; data:Aldus, Pavel, fav, 2014, B, p
[INFO] - 3: vysledek=true; data:Želé, Jakub, fav, 2014, b, P, i
[INFO] - 4: vysledek=true; data:Divoký, René, fav, 2014, B, P, I, něco navíc
[INFO] - 5: vysledek=true; data:Platnost, Karel, fav, 2014, n, p
...

```

- ve třídě `LogovaniVysledku` zakomentujte výpisy do konzole pomocí `System.err`

Kontrola úplnosti řešení:

- adresář `test` z IntelliJ překopírujte někam do svého pomocného adresáře, např. `D:\zzz`
 - dále sem překopírujte obsah adresáře `kontrola/oks03` a soubor `kontrola.bat`
 - dále sem překopírujte obsah adresáře `lib`
 - dále sem překopírujte soubor `data/priklady-oks-03.txt`
 - z IntelliJ sem překopírujte soubor `oks-03-konfigurace.xml`

■ spusťte soubor `kontrola.bat` nebo z něj použijte příslušné příkazy

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

```
-cp apiguardian-api-1.0.0.jar;junit-jupiter-api-5.3.2.jar ...
```

na :

```
-cp apiguardian-api-1.0.0.jar:junit-jupiter-api-5.3.2.jar ...
```

■ porovnejte obsahy souborů `oks-03-log.txt` a `vysledek/vzor-oks-03-log.txt`

- postupujte stejně jako v projektu `oks-01` včetně řešení případného problému s kódováním diakritiky
- pokud nebudou identické, tak ani neodevzdávejte a hledejte příčinu rozdílu

Příprava souborů k odevzdání:

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

```
jar cMf oks-03.jar test *.xml
```

■ výsledkem bude soubor `oks-03.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