

Cíle prvního cvičení

Cílem dnešního cvičení je zprovoznit požadované nástroje a vyzkoušet si práci s nimi.

Instalace požadovaných nástrojů

Pokud pracujete na svém notebooku, nainstalujte si následující nástroje:

- *JDK 7.*
- *NetBeans 7.2*, včetně aplikačního serveru *GlassFish Server Open Source Edition 3.1.2.2* a servlet kontejneru *Apache Tomcat 7.0.27*.
- *TortoiseSVN 1.7.9* (pro MS Windows) nebo *Subversion 1.7.x* (Unix/Linux). Pozor, u TortoiseSVN musíte nainstalovat správnou variantu - pro 64-bitová Windows musíte nainstalovat 64-bitovou verzi TortoiseSVN a pro 32-bitová Windows musíte nainstalovat 32-bitovou verzi TortoiseSVN.
- *Apache Maven 3.0.4.*

Zdroje

- <http://netbeans.org/downloads/index.html>
- <http://tortoisesvn.net/downloads.html>
- <http://maven.apache.org/download.html>

Zprovoznění vývojového prostředí

1. Spusťte NetBeans a zkontrolujte, že máte v záložce services mezi servery zaregistrovaný server GlassFish i Tomcat.
2. Spusťte Tomcat.
3. Založte nový webový projekt a zkuste jej nasadit (*deploy*) na spuštěnou instanci kontejneru Tomcat.
4. Vyzkoušejte, že aplikace funguje.
5. Kroky 2 - 4 zopakujte i pro server Glassfish.
6. Stáhněte si archiv [currency-convector.zip](#) a vyzkoušejte, že jej lze sestavit. To můžete provést z příkazové řádky pomocí příkazu `mvn install`, nebo otevřením v NetBeans a spuštěním operace *Build*.

Práce se systémem Subversion

1. Přečtěte si stručný návod k nástroji Subversion na kore.
2. Založte si lokální repozitory.
3. Založte si lokální adresáře **pepa** a **franta**. V nich budeme simulovat souběžnou práci dvou vývojářů.
4. Jako **pepa** provedte checkout vaší prázdné repozitory.
5. Jako **franta** provedte checkout vaší prázdné repozitory.
6. Jako **pepa** v repozitory založte adresáře `tags`, `branches` a `trunk`.
7. Jako **pepa** do adresáře `trunk` vložte projekt z archivu [currency-convector.zip](#).

8. Jako **pepa** zkontrolujte lokální změny (`svn status`), zkontrolujte, že jde projekt přeložit a že projdou testy (`mvn test`). Pak proveďte **commit**; nezapomeňte připojit smysluplnou zprávu o změně.
9. Jako **franta** proveďte operaci **update** a zkontrolujte, že operace proběhla úspěšně.
10. Jako **franta** otevřete soubor `ExchangeRateTable.java` a doplňte stručný JavaDoc k metodě `setExchangeRate(...)`.
11. Jako **franta** proveďte operaci **commit**.
12. Jako **pepa** otevřete soubor `ExchangeRateTable.java` a doplňte stručný JavaDoc k metodě `getExchangeRate(...)`.
13. Jako **pepa** se pokuste provést operaci **commit**. Všimněte si, že operace selže a že musíte nejdříve provést operaci **update**. Operace `update` detekuje konflikt (pepa i franta změnili stejný soubor), ale konflikt byl automaticky vyřešen. Opět se pokuste provést operaci **commit**, tentokrát by již měla proběhnout bez problému.
14. Jako **franta** proveďte operaci **update**.
15. U obou uživatelů zkontrolujte, že mají oba aktuální verzi souboru `ExchangeRateTable.java`.
16. Jako **franta** otevřete soubor `ExchangeRateTable.java` a opravte první řádek JavaDoc k metodě `setExchangeRate(...)`.
17. Jako **franta** proveďte operaci **commit**.
18. Jako **pepa** otevřete soubor `ExchangeRateTable.java` a opravte první řádek JavaDoc k metodě `setExchangeRate(...)`.
19. Jako **pepa** se pokuste provést operaci **commit**. Všimněte si, že operace selže a že musíte nejdříve provést operaci `update`. Operace `update` detekuje konflikt (pepa i franta změnili stejný soubor), ale tentokrát konflikt není vyřešen automaticky (byl změněn stejný řádek). Vyřešte konflikt ručně a zopakujte operaci **commit**.
20. Založte novou značku `ver_1_0`, která bude obsahovat aktuální verzi projektu. Zároveň založte novou větev `maintain_1_0`, která bude také obsahovat aktuální verzi projektu.0

Zdroje

- <http://kore.fi.muni.cz/wiki/index.php/Subversion>
- <http://svnbook.red-bean.com/en/1.7/svn-book.html>

Maven

1. Vyzkoušejte si, že jde projekt přeložit a nainstalovat do lokální repozitory (`mvn install`). Ověřte si, že se sestavený artefakt do lokální repozitory opravdu nainstaloval.
2. Opravte třídu `CurrencyConvertorTest`, aby místo instance anonymní lokální třídy používal mock objekt vytvořený pomocí knihovny *Mockito*. To samozřejmě předpokládá, že příslušnou závislost doplníte do `pom.xml`.
3. Opravte `pom.xml` tak, abyste se zdrojové kódy překládali překladačem pro verzi jazyka 1.7. Že jste to udělali správně ověříte tak, že ve třídě `CurrencyConvertorTest` nahradíte konstantu `2820.0` za `2_820.0`.