



Testy

Jméno projektu:	nextQuest
Verze:	1.0.0.
Poslední úprava:	17. 12. 2011
Autoři:	Hejduk Jiří, Halouska Matěj, Souček Michal, Obešlo Stanislav, Michálek Tomáš
Jméno klienta:	Macek Ondřej
Kontakt klienta:	macekond@fel.cvut.cz

Obsah:

- Přístup k testování
- LoginControlTest
- QuestControlTest
- StaffControlTest
- AbilityControlTest
- Zhodnocení

Přístup k testování:

Na základě podmnožiny všech možností testového frameworku JUnit, kterou jsme stihli nastudovat, jsme jako nejrozumnější přístup k testování shledali zaměření se na nejdůležitější část aplikační logiky – controlery. Na nich provedené testcase sice neověřují všechny možné kombinace vstupů (což se nám v současné situaci nezdá stěžejní), ale místo toho se soustředí na celkovou funkčnost a interakci jednotlivých částí klienta a serveru. Od testování serveru jsme upustili z důvodu, že jeho metody prakticky výhradně pracují s databází, do které lze vložit téměř cokoliv a záleží na jejím konkrétním typu.

Všechny testy byly zařazeny pod jeden testovací balík (NextQuestGUIClientTestSuite), jenž je volá v něm předdefinovaném pořadí a současně zajišťuje počáteční inicializaci některých objektů, které testované controlery využívají zejména pro účely styku se serverem. Ty jsou inicializovány v prvotní fázi zástupnými mock objekty, ale zároveň je připsán i zakomentovaný příkaz pro jejich inicializaci reálným objektem jako příprava na zprovoznění rozhraní po dokončení jak serverové tak klientské části aplikace. Následuje seznam jednotlivých testovacích tříd:

LoginControlTest:

Tento test je zaměřen na ověření funkčnosti třídy LoginControl.class. V jeho dvou samostatných testcase ověřujeme funkci mechanismu přihlašování se do aplikace prostřednictvím správných resp. špatných identifikačních údajů.

Seznam testcase:

- testSuccessfulLogin
- testUnsuccessfulLogin

QuestControlTest:

Zde testujeme správnou funkci mechanismu odevzdávání a odmítnutí přidělených úkolů realizovaný metodami returnTask a rejectTask. U odmítnutí úkolu, jehož metoda přebírá parametr vyjadřující odůvodnění této akce, opět využíváme dvou separátních testcase pro ověření zda si poradí se správnými i špatnými vstupy.

Seznam testcase:

- testReturnTasktest
- testRejectIncorrect
- testRejectCorrect

StaffControlTest:

V tomto testu ověřujeme veškerou práci s evidencí uživatelů pro účely naší aplikace. V testcase jsou zahrnuty případy realizující přidání osoby, změnu jejích údajů a odebrání osoby ze seznamu zaměstnanců a to kromě posledního případu opět ve variantách se správnými a špatnými vstupními parametry.

Seznam testcase:

- testAddPersonIncorrect
- testAddPersonCorrect
- testChangePersonIncorrect
- testChangePersonCorrect
- testRemovePerson

AbilityControlTest:

Tento test ověřuje funkčnost AbilityControleru. Konkrétně se jedná o přidávání a mazání v aplikaci dostupných schopností, které lze nastavit jednotlivým uživatelům v různých úrovních.

Seznam testcase:

- testAddAbilityIncorrect
- testAddAbilityCorrect
- testRemoveAbility

Zhodnocení:

Vymyslet strukturu testů pro nás nebylo zcela triviální, ale testování nakonec po překonání tohoto problému i problémů s integrací JUnit frameworku do prostředí Netbeans proběhlo víceméně úspěšně (v režimu s mock objekty s úspěšností 100%, v režimu s reálnými objekty k 18. 12. 2011 s úspěšností 46,15%). Alespoň v případě jednoho controleru nám testy pomohly si uvědomit a následně napravit chybný návrh některých jeho metod, které se staraly o věci jim nenáležející a následkem toho přebíraly parametry nevhodného typu.