Masarykova univerzita
Fakulta informatiky

Editor scenárov projektu PerfCake pre NetBeans

Bakalárska práca

Andrej Halaj

Brno, jeseň 2015

Prehlásenie

Prehlasujem, že táto bakalárska práca je mojím pôvodným autorským dielom, ktoré som vypracoval samostatne. Všetky zdroje, pramene a literatúru, ktoré som pri vypracovaní používal alebo z nich čerpal, v práci riadne citujem s uvedením úplného odkazu na príslušný zdroj.

Andrej Halaj

Vedúci práce: RNDr. Adam Rambousek, Ph.D.

Poďakovanie

Ďakujem predovšetkým Ing. Pavlovi Macíkovi za jeho pomoc a trpezlivosť pri práci na doplnku pc4nb. Tiež ďakujem RNDr. Adamovi Rambouskovi, Ph.D. za pomoc a rady pri písaní textu práce.

Zhrnutie

Cieľom tejto práce bolo implementovať grafický editor scenárov PerfCake ako doplnok do vývojového prostredia NetBeans 8.0. Vzhľad doplnku je podľa požiadavku založený na vzhľade už existujúceho doplnku pre vývojové prostredie Eclipse. Doplnok pre NetBeans má názov pc4nb a požiadavkom bolo, aby okrem grafického editoru obsahoval aj sprievodcu vytvorením nového scenára a možnosť spustiť scenár priamo z vývojového prostredia.

Kľúčové slová

PerfCake, NetBeans, doplnok, plug-in, modul, pc4nb

Obsah

1 **Úvod** 1
2 **Nástroj PerfCake** 3
2.1 *Architektúra* 3
2.1.1 Vlastnosti 4
2.1.2 Generátor 5
2.1.3 Beh 6
2.1.4 Odosielateľ 6
2.1.5 Správy 6
2.1.6 Spätná väzba 7
2.1.7 Validácia 8
2.2 *Práca s nástrojom PerfCake* 8
2.3 *Rozšírenie funkcionality* 9
3 **NetBeans** 11
3.1 *Tvorba doplnkov do NetBeans* 11
3.1.1 Nový typ projektu a súboru 12
3.1.2 Editorové okná 12
3.1.3 Sprievodcovia 12
3.1.4 Paleta nástrojov 13
4 **Návrh doplnku a použité technológie** 15
4.1 *Grafický editor scenára* 15
4.2 *Prevod medzi XML a grafickou podobou scenára* 16
4.3 *Vyhľadávanie implementácií komponentov a ich vlastností* . 17
5 **Štruktúra doplnku** 19
5.1 *model* 19
5.2 *project* 20
5.3 *reflect* 20
5.4 *scenario* 20
5.5 *ui* 21
5.5.1 actions 22
5.5.2 palette 22
5.5.3 tableModel 22
5.5.4 wizards a wizards.visuals 23
6 **Ďalšie použité technológie a knižnice** 25
6.1 *Apache Maven* 25
6.2 *MigLayout* 26

7 **Grafické rozhranie doplnku pc4nb a práca s ním** 27
7.1 *Vytvorenie nového scenára* 29
7.2 *Práca so scenárom* 30
7.3 *Spustenie scenára* 32
8 **Záver** 33
Literatúra 35
A **Elektronické prílohy** 37

1 Úvod

Vývoj softvéru dnes často zahrňuje okrem implementácie funkcií, ktoré má program poskytovať, aj jeho testovanie. Kategórií testovania je viacero a pomocou neho môžeme zmierniť niektoré problémy pri vývoji, alebo im dokonca predísť. Okrem splnenia zadaných požiadavkov na funkčnosť alebo správnosť fungovania môžeme požadovať napríklad aj to, aby softvér pracoval spoľahlivo i pri vysokej záťaži. To, ako je program schopný pracovať pri potenciálnej záťaži, ktorej bude počas používania vystavený, overujeme pomocou záťažových testov.

Jedným z nástrojov na testovanie softvéru je PerfCake [7], ktorý umožňuje vykonávať záťažové testy pomocou rôznych rozhraní či protokolov. Základom testu pomocou nástroja PerfCake je súbor typu XML alebo DSL, ktorý sa nazýva scenár. Ten obsahuje nastavenia testu, ktoré si môže používateľ poskladať z menších komponentov rozdelených do niekoľkých kategórií. To zaručuje flexibilitu pri nastavení testu a zároveň znovupoužiteľnosť v ďalších scenároch. Nástroj PerfCake tiež dovoľuje používateľom naprogramovať si svoj vlastný komponent a použiť ho v scenároch. Jedným z nastavení je výška záťaže, ktorá môže byť napríklad preddefinovaná, alebo sa test môže pokúsiť o maximálnu možnú záťaž. Výsledky testu ako je odozva, pamäťová náročnosť, či výskyt únikov v pamäti, môžu byť sprostredkované napríklad pomocou výpisu do konzoly alebo do súboru.

Cieľom tejto práce bolo vytvoriť grafický editor scenárov PerfCake pre vývojové prostredie NetBeans 8.0 [4] a uľahčiť tak používateľom prácu s týmto testovacím nástrojom. Vyvíjaný grafický editor nesie názov pc4nb a je samostatným doplnkom pre spomínané vývojové prostredie, ktorého súčasťou je aj sprievodca vytvorením nového scenára. Doplnkom sa v tomto prípade myslí modul, ktorý rozširuje funkčnosť vývojového prostredia NetBeans. Požiadavkom na doplnok bola tiež možnosť spustiť scenár priamo z prostredia NetBeans. Inšpiráciou k tomuto doplnku bol už existujúci doplnok pre vývojové prostredie Eclipse [9] s názvom PerfClipse [3]. Doplnok pc4nb bol navrhnutý, aby využil jeho jednoduchosť a prehľadnosť.

Na vývoj doplnku pc4nb bol podľa požiadavku použitý nástroj Apache Maven [8] na správu a automatizáciu zostavovania aplikácií. Vytvoril som preto v prostredí NetBeans nový projekt z prednastave-

nej šablóny, ktorá priamo ponúka možnosť podpory nástroja Maven. Šablóna zároveň umožňuje zostavenie balíčku *nbm*, ktorý predstavuje doplnok určený na inštaláciu do vývojového prostredia NetBeans.

Ako ďalší krok som vytvoril základné okno doplnku, ktoré obsahovalo dve záložky. Jedna záložka s názvom *Dizajnér* určená pre grafickú editáciu scenáru, zatiaľ čo druhá, *Zdroj*, zobrazuje scenár v XML podobe. PerfCake síce podporuje formát scenárov aj vo forme súborov typu DSL, no doplnok pc4nb ich tvorbu a editáciu nepodporuje. Popri tom som tiež vytvoril sprievodcu vytvorením nového scenára, v ktorom je možné pomocou grafického rozhrania vytvoriť nový scenár PerfCake. Ďalej som vytvoril panel nástrojov PerfCake, ktorý mal obsahovať jednotlivé komponenty, ako sú správy, generátory alebo validátory. Požiadavkom bolo, aby tieto komponenty mohli byť jednoducho pridané do projektu pomocou ťahu myši.

Po vytvorení grafického rozhrania som pracoval na prevode medzi XML a grafickou reprezentáciou scenára. Scenár bolo potrebné previesť z XML podoby do objektovej štruktúry v jazyku Java. Tieto objekty sú definované nástrojom PerfCake, avšak neobsahujú možnosť informovať o svojich zmenách. Aby sa teda zmeny v objektoch zobrazili používateľovi, musela byť táto funkcionalita v doplnku pc4nb doimplementovaná. To umožňuje, aby sa zmeny v scenári automaticky prejavili v grafickom rozhraní doplnku.

Na záver bolo tiež potrebné implementovať možnosť spustiť a ukončiť beh vytvoreného scenára priamo v prostredí NetBeans, kde je priebeh spusteného scenára možné sledovať v okne Výstup scenára.

2 Nástroj PerfCake

PerfCake [7] je testovací nástroj určený na testovanie výkonu softvérových systémov. Jeho modulárna architektúra umožňuje jednoduché nastavenie testov, ktoré môžu prebiehať cez množstvo protokolov. Jeho cieľom je, aby bol minimalistický, jednoduchý na používanie a aby čo najmenej zaťažoval testovaný systém a mal teda čo najmenší možný dopad na výsledky merania.

2.1 Architektúra

Základom každého testu vykonávaného pomocou nástroja PerfCake je scenár (angl. scenario). Scenár je popísaný súborom typu XML alebo DSL, v ktorom sú definované všetky nastavenia potrebné pre beh testu. Definícia scenára je rozdelená do siedmych kategórií:

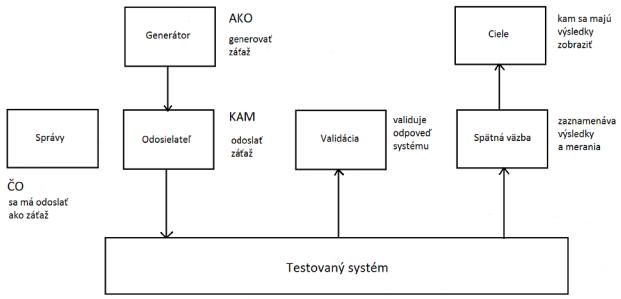
- beh (run)
- generátor (generator)
- odosielateľ (sender)
- správy (messages)
- spätná väzba (reporting)
- validácia (validation)
- vlastnosti (properties)

Listing 2.1: Jednoduchý príklad XML scenára pre PerfCake

```
<scenario xmlns="urn:perfcake:scenario:5.0">
  <run type="time" value="10000">
  <generator class="DefaultMessageGenerator" threads="1" />
  <sender class="DummySender">
      <target>dummyTarget</target>
  </sender>
</scenario>
```

Aby bolo možné scenár spustiť, musia byť definované aspoň komponenty generátor a odosielateľ a tiež nastavenie behu (run). Takto definovaný scenár je schopný generovať a odosielať správy určenému cieľu.

Táto kapitola sa zaoberá popisom jednotlivých kategórií nastavení scenára, no priložené sú len ukážky vo forme XML. Ak nebude uvedené inak, scenárom sa ďalej bude myslieť práve PerfCake XML scenár. Scenármi vo formáte DSL sa táto práca nezaoberá a hotový doplnok pre NetBeans tvorbu a editáciu DSL scenárov nepodporuje. Text práce pracuje s nástrojom PerfCake vo verzii 5.1, no doplnok je naprogramovaný pre vývojovú verziu 6.0-SNAPSHOT [1].



Obr. 2.1: Architektúra nástroja PerfCake

2.1.1 Vlastnosti

Väčšina komponentov PerfCake má okrem základných nastavení aj tzv. vlastnosti (properties), pomocou ktorých je možné ďalej upraviť ich chovanie. Používateľ ich buď nastaví explicitne a vloží svoju vlastnú hodnotu, alebo sa použije preddefinovaná. V prípade použitia vo viacerých komponentoch musí byť daná vlastnosť definovaná v oboch, pretože sa vzťahuje len na jeden konkrétny objekt.

1. viz. 4.3 posledný odstavec

Listing 2.2: Ukážka nastavenia odosielateľa s vlastnosťou

```
<sender class="HTTPSender">
   <target>http://httpbin.org/get</target>
    <property name="method" value="POST" />
</sender>
```

Okrem spomínaných vlastností komponentov scenár obsahuje aj svoju samostatnú kategóriu vlastnosti, ktorá sa neviaže na konkrétny komponent. Tá neobsahuje žiadne nastavenia, ale metadáta o danom scenári.

Listing 2.3: Ukážka nastavenia vlastností scenára

```
<properties>
   <property name="variable" value="10" />
   <property name="scenarioId" value="15022015N124" />
</properties>
```

2.1.2 Generátor

Generátor slúži na generovanie záťaže a využíva objekty typu odosielatel[2] na odosielanie a prijímanie správ. Každý scenár musí obsahovať práve jeden generátor, ktorého implementácia určuje spôsob, akým je generovaná záťaž. Do nastavení spoločných pre každú implementáciu generátora patrí počet vlákien, v ktorých má proces bežať.

Listing 2.4: Ukážka nastavenia generátora

```
<generator class="DefaultMessageGenerator" threads="3">
  <property name="monitoringPeriod" value="2000" />
  <property name="senderTaskQueueSize" value="4000" />
</generator>
```

2. 2.1.4

2.1.3 Beh

Od verzie PerfCake 5.0 je sekcia beh oddelená od generátora. Beh obsahuje typ časovej jednotky a hodnotu určujúcu ako dlho má byť generovaná záťaž.

Listing 2.5: Ukážka nastavenia behu

```
<run type="time" value="5000" />
```

2.1.4 Odosielateľ

Objekt typu odosielateľ musí byť v scenári definovaný taktiež práve jeden a jeho úlohou je odosielať správy do cieľového systému a prijímať odpovede. Jednotlivé typy odosielateľov predstavujú to, akým spôsobom sú správy predávané. Medzi ne patrí napríklad komunikácia pomocou protokolu HTTP či HTTPS alebo odosielanie SQL dotazov prostredníctvom JDBC[3].

Listing 2.6: Ukážka nastavenia odosielateľa

```
<sender class="HttpSender">
   <target>http://httpbin.org</target>
   <property name="method" value="POST" />
</sender>
```

2.1.5 Správy

Sekcia správy obsahuje definíciu jednotlivých správ, ktoré sú použité ako obsah správy (payload) odosielaný odosielateľom pri generovaní záťaže. Obsah správy môže byť uložený v súbore. V tom prípade sa použije nastavenie URI, ktoré špecifikuje, kde sa daná správa nachádza. Pri jednoduchých správach je možné namiesto URI použiť nastavenie obsah (content), kam sa priamo napíše požadovaný obsah správy. Nastavenie počet (multiplicity) určuje, koľkokrát má byť daná správa odoslaná v jednej iterácií testu.

3. Java Database Connectivity

Správa môže tiež obsahovať nastavenie svojich hlavičiek[4] (headers) a vlastností (properties) a odkazy na validátory, ktoré majú odozvu na danú správu validovať.

Listing 2.7: Ukážka nastavenia správ

```
<messages>
   <message uri="message1.txt" multiplicity="4">
      <header name="Accept" value="text/plain" />
      <validatorRef id="1" />
    </message>
   <message content="Hello World!" />
</messages>
```

2.1.6 Spätná väzba

Aby sa výsledky testu zaznamenávali, je potrebné v scenári definovať spätnú väzbu. Tá je definovaná pomocou komponentov typu reportér (reporter). Reportérov môže byť v scenári viacero a každý je zodpovedný za zaznamenávanie len určitých dát, ktoré sú určené typom reportéra. Zaznamenávané dáta môžu byť napríklad odozva testovaného systému alebo množstvo využívanej pamäte.

Samotný reportér sa stará len o zaznamenávanie dát, no aby bolo možné si prijaté výsledky testu prezrieť, je potrebné definovať cieľ (destination) zaznamenávania. Reportér prijaté dáta spracuje a odošle ich na svoje definované ciele. Príkladom cieľu môže byť napríklad súbor.

Listing 2.8: Ukážka nastavenia reportérov

```
<reporting>
   <reporter enabled="true" type="ThroughtputStatsReporter">
      <destination type="Log4jDestination">
         <period type="time" value="1000" />
        </destination>
   </reporter>
</reporting>
```

4. viz. Data Header

Ciele naviac v sebe obsahujú vnorené nastavenia periód (period). Tie sú veľmi podobné nastaveniu beh pre scenár a obsahujú typ časovej jednotky a veľkosť periódy. Periódy určujú, ako často sú dáta zaznamenávané do nadradeného cieľa. Každý cieľ môže mať viacero periód, no minimálne jedna musí byť nakonfigurovaná.

2.1.7 Validácia

Sekcia validácia slúži na kontrolu odpovedí, ktoré dostane komponent odosielateľ späť z testovaného systému. Tu sú definované validátory, ktoré môžu byť v sekcii správy priradené ku správam, ktoré majú validovať. Každá správa môže mať priradených viacero validátorov a každý validátor môže overovať odpovede viacerých správ. Aby bola daná odpoveď pokladaná za validnú, musí prejsť všetkými validátormi.

Listing 2.9: Ukážka nastavenia validácie

```
<validation>
   <validator id="1" class="RegularExpressionValidator" />
</validator>
```

2.2 Práca s nástrojom PerfCake

Aby bolo možne overiť, či je určitý XML súbor validným scenárom, nástroj PerfCake ho pri spustení testu validuje schémou definovanou pomocou jazyka XML Schema. Tá definuje, ako má scenár vyzerať a aké prípadné nastavenia musí alebo môže obsahovať.[5].

Adresár nástroja PerfCake po stiahnutí a rozbalení v sebe obsahuje niekoľko súborov a podadresárov. V adresári *bin* sa nachádzajú spustiteľné súbory pre operačné systémy Windows a Linux. Práca ďalej popisuje prácu s nástrojom PerfCake verzie 5.1 na operačnom systéme Windows.

Po otvorení príkazového riadku a presunutí sa do zložky *bin* v adresári, kde je rozbalený PerfCake, je možné nástroj spustiť pomocou príkazu:

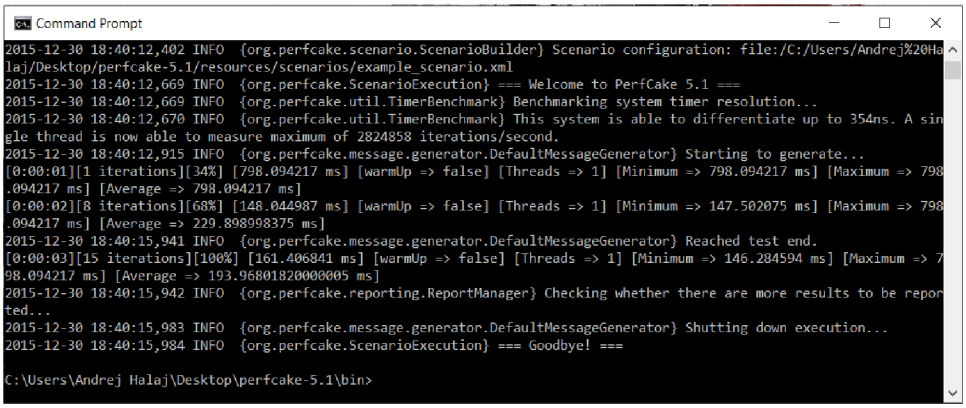
5. viac o XML Schema: http://www.w3.org/standards/xml/schema

```
perfcake
```

Vypíšu sa informácie o tom, ako nástroj používať. Ak chceme spustiť scenár, jednou možnosťou je vložiť súbor s ním do podadresára *resources/scenarios*. Potom ho môžeme spustiť príkazom

```
perfcake -s nazov_scenara
```

Nastavenie *-s* definuje názov scenára, ktorý sa má spustiť. Koncovku scenára je nutné písať len v prípade, že sa v adresári so scenármi nachádza scenár s rovnakým menom vo formátoch XML aj DSL. Ak chceme scenár spustiť z inej zložky ako *resources/scenarios*, je možné pomocou nastavenia *-sd* zvoliť zložku, v ktorej sa má scenár hľadať.



Obr. 2.2: Príklad behu scenára spusteného nástrojom PerfCake.

2.3 Rozšírenie funkcionality

Okrem implementácií komponentov, ktoré sú poskytované spolu s nástrojom PerfCake, si používateľ môže naprogramovať svoje vlastné. Každá kategória komponentov PerfCake má definované vlastné rozhranie v jazyku Java, ktoré má daná komponenta implementovať, aby ju bolo možné následne použiť v scenári. Súbor jar, ktorý obsahuje uživateľskú implementáciu potom stačí vložiť do zložky *lib/plugins* v hlavnom adresári PerfCake.

3 NetBeans

NetBeans je open source integrované vývojové prostredie (ďalej len IDE, Integrated development environment) pre vývoj desktopových, mobilných i webových aplikácií [4]. Ide o oficiálne prostredie na vývoj aplikácií v jazyku Java[1], no podporuje i rad iných jazykov ako napríklad C/C++. Samotné NetBeans IDE je naprogramované v Jave a je teda platformovo nezávislé.

3.1 Tvorba doplnkov do NetBeans

Táto kapitola popsuje spôsoby tvorby doplnkov a rozšírení do prostredia NetBeans, ktoré sú relevantné pre vývoj pc4nb.

Doplnky pre NetBeans IDE sú vytvárané vo forme modulov. Modul je súbor tried v jazyku Java, ktoré ponúkajú určitú funkcionalitu. Samotné vývojové prostredie NetBeans sa skladá z viacerých modulov, ktoré je možné za behu vypínať a zapínať podľa potreby.

Nový modul je možné vytvoriť ako akýkoľvek iný projekt, a to cez položku v menu: Súbor -> Nový Projekt. Tam sa nachádza zložka Moduly NetBeans, v ktorej sa okrem iného nachádza aj nový typ projektu s názvom Modul (Module). Modul je možné vytvoriť samostatne, alebo môže byť súčasťou tzv. Sady modulov (Module Suite). Výhodou sady modulov je, že moduly, ktoré sú jej súčasťou, môžu byť na sebe závislé a taktiež môžu byť spoločne zostavované.

NetBeans už v základnej inštalácií obsahuje v sebe zabudovanú podporu pre nástroj na správu a zostavovanie softvérových projektov Apache Maven. Je teda možné pri vývoji modulov pre NetBeans využiť aj tento nástroj, a to tak, že v ponuke Nový projekt zvolíme zložku Maven. V nej je možné nájsť položku Modul NetBeans, ktorá nám dovolí vytvoriť projekt pre vývoj modulu s podporou nástroja Maven.

Keďže jednou z podmienok kladených na túto prácu bolo, aby bol doplnok pc4nb zostavovaný pomocou Apache Maven, bol pri vývoji zvolený práve tento typ projektu[2].

---

1. Vývoj jazyka Java i prostredia NetBeans je zastrešovaný spoločnosťou Oracle Corporation
2. viac o Maven viz. 6.1

3.1.1 Nový typ projektu a súboru

Prvým krokom v tvorbe doplnku pc4nb bolo vytvorenie nového typu projektu a súboru. Nový typ projektu je potrebné definovať, aby prostredie NetBeans rozoznalo, že ide o projekt doplnku pc4nb. K novému typu projektu je možné vytvoriť aj šablónu, ktorá sa zobrazí v sprievodcovi vytvorenia nového projektu a dovoľuje jednoducho vytvoriť projekt daného typu [5].

Nový typ súboru, ktorý má NetBeans IDE rozoznávať, je možné zadefinovať dvoma spôsobmi: buď pomocou koncovky súboru, alebo pomocou XML menného prostredia [3]. NetBeans však vnútorne nepracuje s koncovkou súboru ani menným prostredím, ale definuje nový typ internetového média (ďalej len MIME) [4]. Tento typ NetBeans používa na to, aby bolo možné priradiť súboru špecifické akcie pre daný typ. Príkladom takejto akcie môže byť napríklad zobrazenie ponuky po pravom kliku myši na súbor v okne projektu, v ktorej sa zobrazia len tie položky, ktoré sú definované pre daný MIME.

3.1.2 Editorové okná

Pri tvorbe nového typu súboru ponúka NetBeans možnosť priradiť danému typu editorové okno, ktoré obsahuje viac záložiek (tzv. multiview), z ktorých každá zobrazuje dáta v súbore iným spôsobom. Pri vytvorení súborového typu pomocou XML menného priestoru NetBeans implicitne vytvorí dve záložky. Prvá zobrazuje XML súbor, druhá grafický panel sady nástrojov Swing [6], v ktorom je možné dáta zobraziť v grafickej podobe napríklad ako graf.

3.1.3 Sprievodcovia

Aby sa dal nový scenár PerfCake jednoducho vytvoriť bez písania XML súboru, bol na tento účel vytvorený sprievodca vytvorením nového scenára. V NetBeans vývojovom prostredí sa nový sprievodca vytvorí pomocou ponuky Súbor -> Nový súbor -> Vývoj modulov -> Sprievodca.

---

3. XML Namespace viac na: http://www.w3schools.com/xml/xml_namespaces.asp
4. MIME type, viac na: http://www.webopedia.com/TERM/M/MIME.html

Medzi nastavenia sprievodcu patrí, o akého sprievodcu ide, či o sprievodcu vytvorením nového súboru, alebo o vlastne definovaného sprievodcu. Ďalej sa dá zvoliť, či bude mať daný sprievodca pevne daný počet krokov, alebo je možné za určitých podmienok niektoré kroky vynechať. Ako posledné je potrebné nastaviť maximálny počet krokov, ktoré sprievodca môže obsahovať. Po zadaní mena sprievodcu a cieľa NetBeans automaticky vytvorí triedy v jazyku Java, ktoré predstavujú novovytvoreného sprievodcu.

Každý krok sprievodcu sa skladá z dvoch tried. Prvá definuje grafické okno, ktoré sa zobrazí používateľovi, druhá logiku daného kroku. V druhej časti je možné definovať napríklad možnosť pokračovať v sprievodcovi ďalej len po vyplnení určitej položky v okne prislúchajúcom tomuto kroku. Druhým dôvodom na využitie logickej časti je, že v nej môžme zvoliť, aké dáta z grafického okna chceme uložiť a spracovať. Môže ísť o dáta z tabuľky, či text vpísaný uživateľom do textového poľa.

NetBeans tiež vytvorí triedu, ktorá obsahuje ukážku kódu, ako s daným sprievodcom pracovať. Tento kód je možné poupraviť a zakomponovať do užívateľského modulu a vyvolať tak sprevodcu keď ho potrebujeme (napríklad po kliku na tlačidlo).

3.1.4 Paleta nástrojov

Paletu nástrojov je do užívateľského modulu v prostredí NetBeans možné implementovať pomocou hierarchie uzlov (node)[1]. Tá je tvorená z objektov, ktoré implementujú rozhranie `AbstractNode`. Najvyššia úroveň predstavuje celú paletu, prostredná úroveň kategórie nástrojov v palete a na najnižšej úrovni sa nachádzajú jednotlivé nástroje.

Ak chceme, aby bolo nástroje možné pridávať do projektu pomocou ťahu myši, musia triedy, ktoré predstavujú uzly týchto nástrojov, obsahovať odkaz na objekt, ktorý sa má vložiť. Tento objekt musí implementovať rozhranie `Transferable`, a cieľ, kam sa má daný objekt pridať, musí mať definovaný svoj obslužný objekt pre prenos (objekt typu `TransferHandler`).

4 Návrh doplnku a použité technológie

Cieľom tejto práce bolo naprogramovať doplnok pre NetBeans IDE, ktorý by umožnil:

- jednoduchú a uživateľsky prívetivú editáciu XML scenárov PerfCake pomocou grafického editoru vstavaného priamo do vývojového prostredia
- tvorbu nového scenára pomocou grafického sprievodcu
- spustenie scenárov priamo z prostredia NetBeans

Návrh doplnku teda bol rozdelený do niekoľkých krokov. Prvým krokom bolo navrhnutie hlavného editorového okna, kde by boli zobrazené nastavenia scenára. Ďalej bolo treba vyriešiť prevod z XML do grafickej podoby a naopak, aby sa zmeny v jednej vrstve prejavili aj v druhej. V treťom kroku bolo potrebné navrhnúť editačné okná pre jednotlivé komponenty. Na záver sa bolo treba uistiť, že projekt s doplnkom pc4nb je rozvrhnutý logicky, keďže projekt má otvorený zdrojový kód a predpokladá sa ďalší vývoj aj od iných prispievateľov.

4.1 Grafický editor scenára

Základnou myšlienkou práce bolo predovšetkým uľahčiť prácu s nástrojom PerfCake, ktorý pre beh testov využíva nastavenia zapísané v scenároch. Textová editácia XML scenárov je zdĺhavá a núti používateľov pamätať si všetky nastavenia nutné pre každú sekciu či komponent, prípadne si ich používateľ musí vyhľadať v dokumentácií.

Návrh doplnku pc4nb teda mal dbať na to, aby boli dostupné implementácie komponentov zobrazené pri ich vytváraní či editovaní. Taktiež mali byť zobrazené aj možnosti pre nastavenia vlastností komponentov. Ak si teda používateľ zvolí daný typ komponentu, v tabuľke vlastností sa majú zobraziť tie vlastnosti, ktoré komponent dovoľuje nastaviť. To by výrazne zjednodušilo prácu s vlastnosťami. Používateľ by ich nemusel vyhľadávať v dokumentácií a ručne zadávať.

V dobe písania tejto práce už existoval doplnok pre Eclipse IDE s názvom PerfClipse[3]. Grafické rozhranie doplnku PerfClipse sa

stalo inšpiráciou aj pre pc4nb. Návrh hlavného editorového okna teda využil jednoduchosť a prehľadnosť doplnku pre Eclipse a vo výsledku sú si v tomto ohľade doplnky podobné.

Editorové okno je teda rozdelené podľa sekcií scenára PerfCake. Tieto sekcie môžu obsahovať svoje vnorené komponenty, no inak sú zobrazené len základné informácie o nich, aby sa zachovala prehľadnosť. Vedľa editorového okna sa tiež nachádza paleta, z ktorej sa dajú pomocou ťahu myši komponenty pridávať do scenára. Editorové okno je možné prepnúť medzi grafickým a XML zobrazením scenára.

4.2 Prevod medzi XML a grafickou podobou scenára

Hlavnou úlohou pri vytváraní doplnku pc4nb bolo vyriešiť transformáciu XML scenára do grafickej podoby a naopak. Navrhnuté riešenie používa grafické objeky v editore na zobrazenie scenára, z ktorých každý má svoj model. V modeli sú uchovávané informácie o danom komponente PerfCake, ktorý je jeho grafickým objektom zobrazený používateľovi. To znamená, že zobrazované dáta sú oddelené od ich grafickej reprezentácie a model sa niako nestará o to, ako je používateľovi zobrazovaný.

Na prevod medzi XML a objektovou štruktúrov v jazyku Java existuje nástroj JAXB[1]. Pc4nb využíva JAXB na prevod medzi XML scenárom a objektami v Jave, ktorých definícia je súčasťou nástroja PerfCake. JAXB využíva na validáciu scenára schému definovanú pomocou jazyka XML Schema, ktorú používa aj samotný nástroj PerfCake. Vytvorené objekty tvoria hierarchickú štruktúru scenára PerfCake. Objekty na nižšej vrstve predstavujú jednotlivé komponenty ako sú reportéry či validátory. Stredná vrstva predstavuje sekcie scenára, teda validácia či správy, a na vrchu štruktúry sa nachádza scenár.

Takto vytvorené objekty však ešte nie je možné použiť ako základ pre grafický editor scenárov, kedže neobsahujú možnosť informovať o svojich zmenách. To znamená, že prípadné zmeny v scenári by nebolo možné zachytiť, spracovať a okamžite zobraziť v editore doplnku pc4nb. Bolo teda potrebné zaviesť vlastné modely objektov PerfCake, ktoré by túto funkcionalitu obsahovali.

1. Java Architecture for XML Binding

Doplnok pc4nb definuje svoje vlastné typy objektov, ktoré v sebe pomocou kompozície obsahujú objekty PerfCake, no naviac pri každej zmene informujú grafické rozhranie. Tým je zaistené, že sa zmeny v objektovej štruktúre scenára okamžite zobrazia uživateľovi.

Pc4nb zatiaľ neobsahuje možnosť zachovávať komentáre v XML súbore scenára. Ak teda scenár pôvodne nejaké komentáre obsahoval, po prepnutí do grafického zobrazenia a naspäť do XML scenár komentáre obsahovať nebude.

4.3 Vyhľadávanie implementácií komponentov a ich vlastností

Návrh doplnku počíta s tým, že ponúkané typy komponentov a ich vlastnosti budú zobrazené používateľovi. Bolo teda potrebné nejakým spôsobom z nástroja PerfCake získať potrebné informácie o komponentoch. Pc4nb na túto úlohu používa reflexiu.

"Reflexia v jazyku Java umožňuje za behu programu získať informácie o rozhraniach, triedach a ich poliach (field) a metódach. Pomocou reflexie je možné tiež vytvárať nové inštancie objektov, volať ich funkcie alebo nastavovať hodnoty polí."[2] Na vyhľadanie všetkých typov komponentu z danej kategórie scenára teda stačí získať všetky implementácie rozhrania[2] pre túto kategóriu. Každá z kategórií scenára má totiž definované rozhranie v Jave, ktoré musia komponenty patriace do danej kategórie implementovať. Na nájdenie implementácií komponentov je použitá knižnica *Reflections*[3].

Získanie vlastností komponentov už je dostiahnuté pomocou štandardnej knižnice jazyka Java s názvom *Java Reflections API*. Triedy komponentov PerfCake momentálne neobsahujú indície, akými by mohli byť napríklad anotácie, ktoré z polí v danom komponente predstavujú vlastnosti. Doplnok Pc4nb preto musí vyfiltrovať potrebné polia z kódu.

Po získaní týchto polí a teda z nich aj názvov vlastností je pomocou reflexie vytvorený nový objekt daného typu a z neho pomocou zavolania metód získané prednastavené hodnoty. Tieto hodnoty sú po-

2. V tomto prípade sa rozhraním myslí rozhranie v jazyku Java
3. Reflections: https://github.com/ronmamo/reflections

tom použité v grafickom zobrazení vlastností, aby bolo možné rozlíšiť uživateľsky nastavené hodnoty od tých prednastavených.

Poslednou úlohou je rozlíšenie, ktoré vlastnosti sú pre konkrétny typ komponentu povinné. PerfCake pre túto potrebu v sebe obsahuje anotáciu `MandatoryProperty`, ktorá je priradená každému poľu predstavujúcemu povinnú vlastnosť. Tie je tiež možné získať pomocou reflexie a v tabuľke vlastností ich odlíšiť od nepovinných.

Vo fáze projektu, kedy bola táto funkcionalita implementovaná, používal doplnok pc4nb verziu PerfCake 5.1. Tá už síce obsahovala potrebné anotácie pre označenie povinných nastavení vlastností, no tie neboli nastavené tak, aby boli prístupné za behu programu. Nebolo teda možné pomocou reflexie tieto anotácie nájsť a rozlíšiť, či ide o povinnú vlastnosť. Preto pc4nb používa vývojovú verziu PerfCake 6.0-SNAPSHOT, kde už sú anotácie prístupné pomocou reflexie.

5 Štruktúra doplnku

V tejto kapitole je popísaná štruktúra doplnku pc4nb ako projektu v Jave. Obsahuje teda rozpis jednotlivých balíkov a tiež popis, čo sa v nich nachádza a k akému účelu slúžia. Projekt obsahuje päť hlavných balíkov:

- model (org.perfcake.pc4nb.model)
- project (org.perfcake.pc4nb.project)
- reflect (org.perfcake.pc4nb.reflect)
- scenario (org.perfcake.pc4nb.scenario)
- ui (org.perfcake.pc4nb.ui)

5.1 model

Doplnok pc4nb je navrhnutý tak, že dáta jednotlivých komponentov PerfCake sú uložené v jednom objekte nazývanom model a o ich zobrazenie sa stará iný objekt s názvom pohľad (view).

Nástroj PerfCake už obsahuje v sebe definíciu tried, ktoré predstavujú časti scenára ako sú reportéry alebo generátory. V balíku *model* sa nachádzajú triedy, ktoré v sebe obsahujú inštancie týchto objektov, no rozširujú ich funkcionalitu tak, aby grafické rozhranie vedelo o prípadných zmenách. Obsahujú vlastné metódy, ktoré menia dáta v objektoch nástroja PerfCake a keď nastane zmena, informujú objekty predstavujúce pohľad na ne, kde daná zmena nastala.

Keďže objekty, ktoré predstavujú modely doplnku pc4nb v sebe priamo obsahujú objekty nástroja PerfCake, je možné k ním bez problémov pristupovať. Ak však máme k dispozícií len referenciu na objekt PerfCake a chceme získať prístup k jeho pc4nb modelu, je nutné tento vzťah niekam uložiť. Na to slúži objekt typu ModelMap, ktorý je implementovaný ako jedináčik[1]. V ňom sú uložené objekty PerfCake a k ním prislúchajúce modely. To je potrebné vo chvíli, keď meníme

1. Podľa návrhového vzoru jedináčik - Singleton

údaje v objekte nástroja PerfCake, no chceme, aby sa zmena prejavila v jeho pohľade a musíme teda použiť metódy príslušného modelu.

5.2 project

Balík *project* obsahuje triedy definujúce projekt typu PerfCake doplnku pc4nb a tiež sprievodcu vytvorením nového projektu.

5.3 reflect

Tento balík obsahuje dve triedy. Jedna je určená na to, aby pre zadaný typ komponentu PerfCake našla všetky jeho implementácie a vrátila príslušné triedy. Jej funkcionalitu využívajú sprievodcovia vytvorenia a editácie nového komponentu a paleta nástrojov, aby bolo možné zobraziť všetky implementácie z danej kategórie.

Druhá trieda slúži na vyhľadanie všetkých vlastností zadaného komponentu, ktoré sú potom zobrazené v tabuľke vlastností tohto komponentu. Okrem názvov vlastností sú vrátené aj ich implicitné hodnoty a tiež to, ktoré vlastnosti sú pre danú komponentu povinné. To umožňuje v tabuľke vlastností zvýrazniť povinné vlastnosti a tiež rozlíšiť, či sú nastavené hodnoty implicitné alebo nastavené používateľom.

5.4 scenario

V balíku *scenario* sa nachádzajú jednak triedy, ktoré definujú nový typ súboru predstavujúci scenár PerfCake, ale aj triedy obsluhujúce spúšťanie, zastavenie a prevod medzi XML a objektovou formou scenára.

Trieda `ScenarioManager` obsahuje dve metódy, ktoré kontrolujú beh scenára: `runScenario` a `stopScenario`. Metóda `createModel` zo zadanej URL načíta scenár a prevedie ho pomocou JAXB do objektovej štruktúry nástroja PerfCake. Posledná metóda `createXML` slúži na prevod z opačného smeru, teda z objektov v jazyku Java do XML.

Triedy `RunScenario` a `StopScenario` definujú akcie, ktorými je možné pristupovať k metódam `runScenario` a `stopScenario` objektu typu `ScenarioManager` z vývojového prostredia NetBeans. Tieto akcie sa zobrazia pri pravom kliku myši na scenár PerfCake v okne *Projekty*.

Ďalšou triedou v tomto balíku je `ScenarioRunWorker`, ktorá dedí z triedy `SwingWorker`[2]. Inštancie tohoto typu slúžia na beh dlhodobých úloh. Aby tieto úlohy neblokovali užívateľské prostredie, je nutné ich spustiť v osobitnom vlákne. SwingWorker sám vlákno vyberie a spustí v ňom zadanú úlohu.

5.5 ui

V balíku *ui* sú definované triedy, ktoré slúžia na grafické zobrazenie scenára a prácu s ním. Tento balík obsahuje veľké množstvo tried a je preto ďalej rozdelený do podbalíkov. V rámci jednoduchosti je uvedený len lokálny názov podbalíka a nie jeho celý názov.

- actions
- palette
- tableModel
- wizards
- wizards.visuals

V hlavnom balíku *ui* sú definované triedy, ktoré predstavujú časti scenára PerfCake v jeho grafickom zobrazení. Všetky pohľady majú spoločného predka s názvom `PC4NBView`, ktorý dedí z triedy `JPanel` nástroja Swing. `PC4NBView` obsahuje nastavenia spoločné pre jeho potomkov ako sú veľkosť, hlavička či farba okraja.

Obsahuje tiež obslužné metódy a vnorené triedy, ktoré umožňujú daný pohľad zvoliť a zmeniť farbu a hrúbku okraja pri jeho zvolení, prípadne vrátiť okraj do pôvodnej podoby pri zvolení iného pohľadu.

Triedy dediace z `PC4NBView` slúžia ako pohľady kategórií nastavení scenára PerfCake ako sú spätná väzba alebo validácia. Pre zobrazenie jednotlivých komponentov, či správ v týchto pohľadoch bol vytvorený potomok dediaci z `PC4NBView` s názvom `SecondLevelView`. Triedy komponentov, správ a vlastností, ktoré sú na najnižšej úrovni hierarchie dedia z triedy `SecondLevelView`, ktorá len upravuje rozmery.

2. viac: https://docs.oracle.com/javase/tutorial/uiswing/concurrency/worker.html

5.5.1 actions

Tu sú obsiahnuté triedy s akciami, ktoré sú používané na pridávanie, editáciu a odoberanie komponentov. Na vrchu hierarchie je abstraktná trieda, ktorá obsahuje kód pre prípadné zobrazenie grafického panelu, ak to akcia vyžaduje. Napríklad pri pridávaní reportéru sa objaví prednastavený panel pre reportér, kde je možné nastavenia upraviť a reportér pridať kliknutím na tlačidlo *Dokončiť*. Abstraktná trieda potom spustí abstraktnú obslužnú metódu, ktorá vykoná potrebné zmeny v modeli komponentu. Ostatné triedy implementujú, aký konkrétny panel sa má zobraziť počas akcie a čo má prebehnúť v obslužnej metóde. Akcie je potom možné v kóde jednoducho inštancovať a spustiť pomocou metódy `execute`.

5.5.2 palette

V podbalíku palette sú triedy, ktoré implementujú paletu komponentov PerfCake, ktorá sa zobrazí v grafickom editore[3].

5.5.3 tableModel

Tento podbalík zahrňuje triedy, ktoré slúžia na zobrazenie komponentov v tabuľkách. Tie majú ako spoločného predka `AbstractTableModel`, ktorý je súčasťou jazyka Java. Tabuľky v nástroji Swing majú svoj model, ktorý predstavuje zobrazované dáta v tabuľke a možnosť práce s nimi. `TableModel` udáva počet riadkov, stĺpcov, ktoré bunky sú editovateľné a pod. Imlementácie tohto modelu v doplnku pc4nb v sebe uchovávajú list objektov, ktoré predstavujú komponenty PerfCake. Tie je možné dynamicky pridať, odobrať a meniť. Teda napríklad `ReportersTableModel` v sebe obsahuje list objektov typu `Reporter`. Vlastné implementácie metód z nadtriedy v každom tabuľkovom modeli potom využívajú informácie z týchto objektov a zabezpečujú, aby sa dáta v tabuľke správne zobrazovali a aby ich bolo možné jednoducho upravovať.

3. viz. 3.1.4

5.5.4 wizards a wizards.visuals

Tieto podbalíky v sebe obsahujú triedy implementujúce sprievodcu nového scenára. Sprievodca je zložený z niekoľkých krokov a každý krok je rozdelený na dve triedy. Jedna predstavuje samotný grafický panel, kde sú zobrazené textové polia či tabuľky. Druhá slúži na obsluhu jej prislúchajúcemu panelu. Pc4nb túto oblužnú triedu používa na uloženie informácií z grafického panelu, aby ich bolo možne po skončení sprievodcu spracovať. Tiež je tu nastavená logika zobrazovania výstražných správ v prípade nenastavených povinných hodnôt. Ak teda napríklad u správy nenastavíme URI ani obsah, vypíše sa správa používateľovi a ten nemôže v sprievodcovi pokračovať, kým jednu z nich nenastaví.

Triedy implementujúce sprievodcu nového scenára sú využívané aj pri vytváraní, či editácií komponentov už vo vytvorenom scenári. Akcie z balíka *actions*, ktoré tieto činnosti umožňujú, v sebe vytvárajú sprievodcu obsahujúceho práve jeden panel pre konkrétny komponent. Týmto spôsobom sú použité rovnaké panely pre prácu či už na novom alebo existujúcom scenári.

6 Ďalšie použité technológie a knižnice

6.1 Apache Maven

Jedným z požiadavkov na doplnok pc4nb bolo, aby bol automaticky zostavovaný pomocou nástroja Apache Maven [8]. Maven je nástroj na správu softvérových projektov v programovacom jazyku Java. Predstavuje jednotný systém zostavovania projektov a správy závislostí pre projekty v jazyku Java a je dnes používaný vo veľkom množstve aplikácií.

Maven je založený na koncepte projektového objektového modelu (POM - Project Object Model), ktorý obsahuje metadáta o danom projekte a jeho zostavení. Okrem iného obsahuje projektový objektový model aj závislosti projektu, teda externé knižnice či nástroje, ktoré sú v ňom využívané. Ak Maven nenájde potrebné závislosti lokálne, je schopný ich automaticky stiahnuť z externého repozitára. Pomocou správy závislostí bolo možné jednoducho pridať funkcionalitu nástroja PerfCake do projektu pc4nb.

Doplnok momentálne používa verziu PerfCake 6.0-SNAPSHOT, ktorá nie je v externom repozitári. Je teda potrebné stiahnuť zdrojové kódy zo serveru GitHub. Z nich je potrebné nainštalovať nástroj do lokálneho repozitára pomocou príkazu:

```
mvn install
```

Po oficálnom vydaní verzie 6.0 a zavedení do externého repozitára stačí zmeniť v súbore *pom.xml* verziu PerfCake na 6.0 a už nebude nutné tento krok vykonávať. Maven si z externého repozitára PerfCake stiahne sám a použije ho pri zostavení doplnku. Doplnok pc4nb samotný je zostavený príkazom:

```
mvn package
```

Tento príkaz vytvorí balíček typu *nbm* (NetBeans Module). Balíček *nbm* sa dá v NetBeans potom nainštalovať cez ponuku Nástroje -> Doplnky -> Stiahnuté -> Pridať doplnky.

Práca obsahuje v elektronickej prílohe už zostavený balíček *nbm*, ktorý je možné hneď nainštalovať bez nutnosti doplnok zostaviť[1].

1. viz. appendix A

6.2 MigLayout

Grafické komponenty v nástrojovej sade Swing umožňujú nastaviť to, akým spôsobom sú rozložené ich vnorené grafické komponenty (ďalej len layout). Swing samotný obsahuje niektoľko implementácií layoutu, no pc4nb využíva dodatočnú implementáciu s názvom *MigLayout*[2].

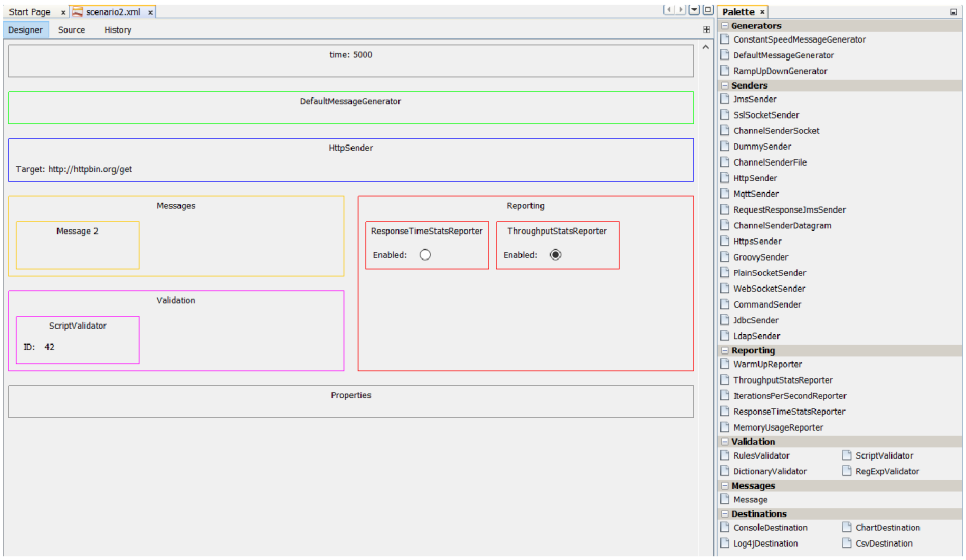
MigLayout vylepšuje a uľahčuje prácu pri návrhu layoutov. Obsahuje viac nastavení, ktoré umožňujú layout presnejšie a jednoduchšie zadefinovať. Príkladom je rozloženie sekcií v editorovom okne pc4nb, ktoré sú rozložené do mriežky, pričom niektoré sekcie sú roztiahnuté cez viac ako jednu bunku mriežky. Roztiahnutie jednotlivých sekcií v mriežke, medzery medzi nimi a celkový layout bol vytvorený pomocou nástroja MigLayout.

2. MigLayout: http://www.miglayout.com

7 Grafické rozhranie doplnku pc4nb a práca s ním

Grafické rozhranie doplnku pc4nb je inšpirované rozhraním doplnku PerfClipse pre vývojové prostredie Eclipse. Je jednoduché a navrhnuté tak, aby boli jednotlivé kategórie komponentov z XML scenára viditeÍné aj v grafickej reprezentácií.

Rozhranie doplnku pc4nb sa skladá z hlavného okna editora v strede a palety nástrojov na pravej strane (viz. obrázok 7.2). Doplnok tiež využíva okno *Projekty* na ľavej strane (nie je zobrazené na obrázku), ktoré je súčasťou NetBeans. V ňom sú zobrazované aj projekty PerfCake. Pri spustení scenára sa tiež zobrazí okno *Výstup scenára*. Hlavné okno editora obsahuje dve záložky. Jednou z nich je *Dizajnér*, ktorá zobrazuje scenár v grafickej podobe. V druhej záložke s názvom *Zdroj* je scenár zobrazený ako XML. Paleta komponentov je viditeľná len ak je grafický editor nastavený na záložke *Dizajnér*. V záložke *Zdroj* nie je možné paletu použiť.



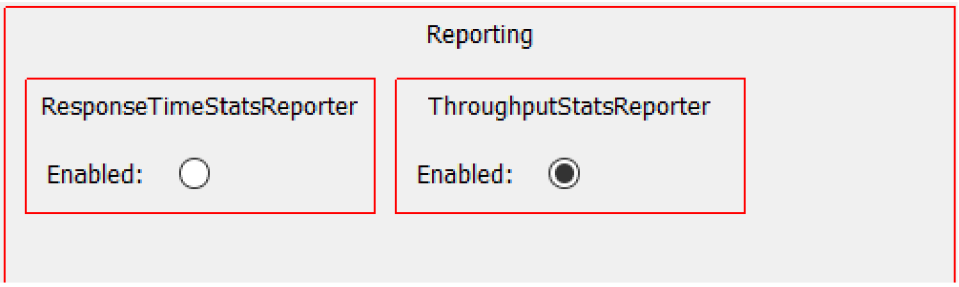
Obr. 7.1: Grafické prostredie doplnku pc4nb.

Grafický editor je teda rozvrhnutý na sekcie predstavujúce sekcie scenára PerfCake. Tieto sekcie sú zobrazené v hlavnom okne editora ako obdĺžniky v mriežke 6x2. Sekcie, ktoré v sebe obsahujú vnorené komponenty, ako napríklad *validácia* obsahuje validátory, zobrazujú taktiež vo vnútri svojich potomkov. Každá sekcia je spolu s jej potomkami od ostatných sekcií farebne odlíšená. Teda napríklad *validácia* a validátory, ktoré obsahuje, majú všetky spoločnú fialovú farbu. Výška obdĺžnikov je nastavená tak, aby sa do nej zmestili všetky vnorené komponenty, zatiaľ čo šírka závisí na veľkosti okna editoru.

Každý obdĺžnik obsahuje hlavičku a prípadne svoje vnorené komponenty alebo doplňujúce informácie pre používateľa. Hlavička obdĺžnika obsahuje to, čo daný obdĺžnik definuje:

- názov sekcie pre sekcie scenára
- typ časovej jednotky a dĺžka behu pre sekciu beh
- URI správy alebo jej obsah pre jednotlivé správy
- názov vlastnosti pre vlastnosti scenára
- typ komponentu pre ostatné komponenty

Doplňujúce informácie sú napríklad zobrazenie identifikačného reťazca (ID) pre validátory, alebo to, či je reportér zapnutý.



Obr. 7.2: Nastavenie reportérov, kde ľavý je vypnutý, pravý zapnutý.

Sekcia beh (run) bola vo verzii PerfCake 5.0 oddelená od generátora a je teraz samostatnou sekciou v scenári. Grafické rozhranie doplnku

pc4nb preto taktiež obsahuje samostatnú kategóriu pre beh na vrchu okna.

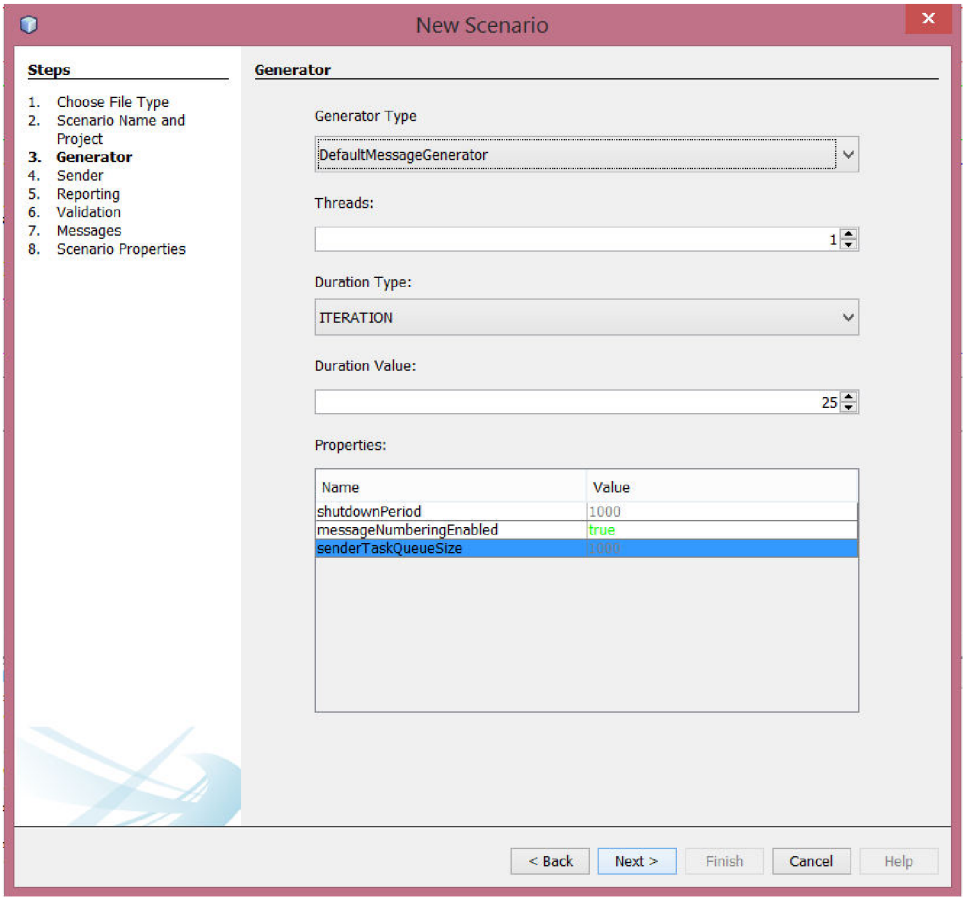
Ostatné vnorené nastavenia, ako sú napríklad nastavenia vlastností u komponentov alebo nastavenia cieľov u reportérov, sa v grafickom editore nezobrazujú, aby bolo prostredie jednoduché a prehľadné. Všetky potrebné informácie o nastaveniach sú zobrazené po otvorení editačného okna daného komponentu či správy, prípadne sú dostupné cez záložku *Zdroj* v XML podobe.

7.1 Vytvorenie nového scenára

Nový scenár PerfCake je do projektu možné pridať pomocou vyvolania kontextového menu po pravom kliku myši na PerfCake projekt a zvolení Nový -> Scenár, prípadne pomocou hlavného menu Súbor -> Nový Súbor -> PerfCake -> Scenár.

Po tomto kroku sa zobrazí na obrazovke sprievodca vytvorením nového scenára. Tento sprievodca obsahuje okno, v ktorom si používateľ zvolí názov scenára a do ktorého projektu scenár patrí. Ďalšie okná predstavujú už jednotlivé sekcie nástroja PerfCake, kde je možné vytvoriť komponeny a zvoliť počiatočné nastavenia. Okná pre nastavenia komponentov použité v sprievodcovi vytvorením scenára sú použité aj pri ich editácií už vo vytvorenom scenári.

Na obrázku 7.3 je možné vidieť aj tabuľku s vlastnosťami, kde sú ponúkané vlastnosti pre daný typ komponentu (v tomto prípade `DefaultMessageGenerator`). Názov vlastnosti je zobrazený čiernou farbou v prípade, že nastavenie danej vlastnosti nie je povinné. Ak je niektorá z nich povinná, je označená červenou farbou. Hodnota vlastnosti je zobrazená sivou farbou, ak ide o prednastavenú hodnotu. V prípade, že je hodnota odlišná od tej prednastavenej, je zobrazená zelenou farbou. To umožňuje používateľovi rozoznať, ktoré vlastnosti nastavil a ktoré budú zobrazené v XML scenári. Vlastnosti s prednastavenými hodnotami v XML scenári nie sú zobrazené, pretože to nie je potrebné a scenár by bol zbytočne dlhý.



Obr. 7.3: Okno generátora v sprievodcovi vytvorením nového scenára.

7.2 Práca so scenárom

Grafický editor scenára v doplnku pc4nb dovoľuje komponenty pridávať, editovať a odoberať. Komponent je možné pridať dvoma spôsobmi.

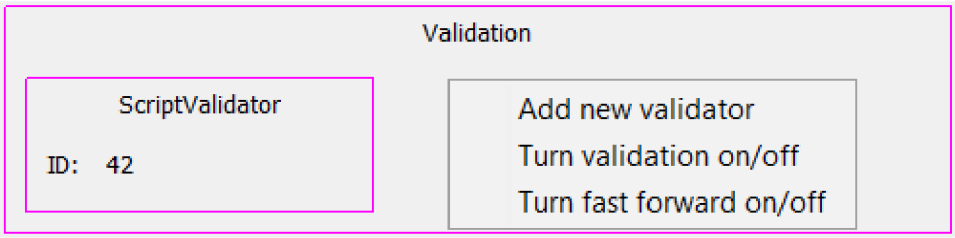
Prvým spôsobom je použitie palety komponentov, z ktorého sa dá pomocou kliku myši zvoliť požadovaný nástroj a pomocou ťahu myši presunúť do scenára. Komponenty je možné vložiť len do tej sekcie scenára, do ktorej patria. Napríklad reportér je možné vložiť len do obdĺžnika, ktorý predstavuje spätnú väzbu (reporting).

Druhý spôsob pridania komponentu je využitie menu, ktoré sa zobrazí po pravom kliknutí myši na niektorú zo sekcií scenára, ktorá vnorené komponenty môže obsahovať. Toto menu je špecifické pre každú takú sekciu a je teda aj týmto spôsobom možné komponenty pridávať len do príslušnej sekcie.

Na editáciu komponentov slúži editačné okno, ktoré sa otvorí po dvojkliku na komponent. Toto okno je taktiež špecifické pre danú kategóriu komponentu. Pri jednoduchom kliku na komponent sa jeho farba okraja zmení na čiernu, čo znamená, že je označený. Komponent je po označení možné vymazať pomocou klávesy **delete**.

Pravím kliknutím na komponent sa otvorí kontextové menu, kde je taktiež možné zvoliť editáciu alebo vymazanie. V kontextovom menu sekcie validácia a komponentov typu reportér je tiež možné danú sekciu či komponent zapnúť alebo vypnúť. Validácia naviac ešte obsahuje položku, ktorá umožňuje zapnúť alebo vypnúť nastavenie *rýchlo dopredu* (fast forward).

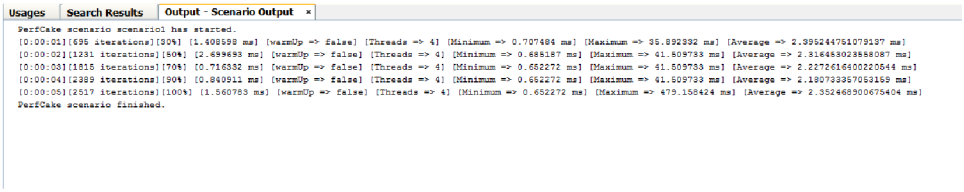
Zmeny z XML podoby scenára sa v grafickej podobe zobrazia po uložení. Zmeny z grafickej podoby sa v XML podobe prejavia po prepnutí na záložku *Zdroj*.



Obr. 7.4: Kontextové menu pre sekciu validácia.

7.3 Spustenie scenára

Doplnok pc4nb umožňuje spustenie scenára priamo v prostredí NetBeans pomocou kontextového menu scenára, ktoré sa otvorí po pravom kliku myši na scenár v okne *Projekty*. V tomto menu sú na výber položky *Spustiť scenár* a *Zastaviť scenár*. Scenár je možné spustiť len ak nebeží a zastaviť len počas behu. Po spustení scenára sa v popredí zobrazí okno *Výstup - Výstup scenára*, kde je možné sledovať jeho beh.



Obr. 7.5: Výstup scenára v doplnku pc4nb.

8 Záver

Cieľom práce bolo vytvoriť editor scenárov pre nástroj na záťažové testovanie PerfCake ako doplnok do vývojového prostredia NetBeans 8.0. Doplnok má názov pc4nb a umožňuje jednoduchú a užívateľsky prívetivú editáciu XML scenárov PerfCake. Okrem editácie dovoľuje vytvoriť nový scenár pomocou grafického sprievodcu a spustenie scenára priamo v prostredí NetBeans, kde je možné sledovať aj výstup scenára.

Grafický sprievodca doplnku pc4nb obsahuje jednotlivé sekcie scenára PerfCake. V nich sa dajú priamo pridať nové komponenty a vytvorený scenár už teda môže obsahovať potrebné nastavenia.

Editácia scenára je umožnená pomocou editačného okna, ktoré obsahuje dve záložky: *Dizajnér* a *Zdroj*. Záložka *Dizajnér* predstavuje grafický editor scenára, *Zdroj* slúži na editáciu scenára prostredníctvom XML editora.

V editačných oknách komponentov, ktoré môžu obsahovať nastavenie vlastností, je priamo zobrazená tabuľka s možnými vlastnosťami pre zvolenú implementáciu komponentu. Názvy vlastností sú farebne rozlíšené podľa toho, či ide o povinnú alebo nepovinnú vlastnosť. Používateľom nastavené hodnoty vlastností sú tiež farebne rozlíšené od predvolených.

Doplnok pc4nb obsahuje paletu komponentov PerfCake rozdelených podľa sekcií scenára. Z nej je možné pomocou ťahu myši pridávať nové komponenty do grafického editora, v prípade generátora a odosielateľa meniť ich typ.

Pc4nb nie je schopný detekovať uživateľské komponenty. Nie sú teda ponúkané ani v palete komponentov, ani ich pri vytváraní či editácií.

Program pc4nb by ďalej mohol byť rozšírený práve o podporu uživateľských implementácií komponent. Ťiež by doplnok mohol obsahovať podporu pre zachovávanie komentárov v XML scenári.

Literatúra

1. BÖCK, Heiko. *Platforma NetBeans: podrobný průvodce programátora*. Brno: Computer Press, 2010.
2. JENKOV, Jakob. *Java Reflection Tutorial* [online]. [cit. 2015-12-30]. Dostupný z WWW: ⟨`http://tutorials.jenkov.com/java-reflection/index.html`⟩.
3. KNETL, Jakub. *PerfClipse: Eclipse plugin which brings support for PerfCake* [online]. [cit. 2015-04-10]. Dostupný z WWW: ⟨`https://github.com/PerfCake/PerfClipse`⟩.
4. *NetBeans IDE: The Smarter and Faster Way to Code* [online]. [cit. 2014-12-08]. Dostupný z WWW: ⟨`https://netbeans.org`⟩.
5. ORACLE CORPORATION. *NetBeans Platform Learning Trail* [online]. [cit. 2015-05-12]. Dostupný z WWW: ⟨`https://netbeans.org/features/platform/all-docs.html`⟩.
6. ORACLE CORPORATION. *Trail: Creating a GUI With JFC/Swing* [online]. 2015 [cit. 2015-05-12]. Dostupný z WWW: ⟨`http://docs.oracle.com/javase/tutorial/uiswing`⟩.
7. *PerfCake: A Lightweight Performance Testing Framework* [online]. [cit. 2014-11-07]. Dostupný z WWW: ⟨`https://perfcake.org`⟩.
8. THE APACHE SOFTWARE FOUNDATION. *Apache Maven* [online]. [cit. 2015-02-08]. Dostupný z WWW: ⟨`http://maven.apache.org`⟩.
9. THE ECLIPSE FOUNDATION. *Eclipse* [online]. 2014 [cit. 2014-12-08]. Dostupný z WWW: ⟨`https://eclipse.org`⟩.

A Elektronické prílohy

Práca vložená do systému IS v prílohe obsahuje:

- zdrojové kódy práce
- zostavený balíček *nbm*, ktorý je možné nainštalovať ako doplnok do NetBeans