

# Aspektovo riadená konfigurácia vlastností softvérového radu

Generovanie variantov produktov pomáha vytvoriť produkty z danej domény efektívnejšie a lacnejšie. Nevzniká tak iba jeden produkt, ale ľahko adaptovateľné produkty pre rôzne prostredia, jazyky a ďalšie požiadavky (Beuche a Dalgarno 2006). Derivovať produkty prispôbené požadovanej variabilite možno za pomoci aspektovo orientovaného prístupu, hlavne jazyka AspectJ<sup>1</sup> (Young a Math, 1999), ktorého možnosti by sme chceli pri tejto úlohe preskúmať. Vyberáme použitie jazyka AspectJ, pretože umožňuje oddeliť záležitosti ako je logovanie, kešovanie, poolovanie, ale hlavne je možné vyradiť z vykonávania vybrané aspekty, respektíve isté vlastnosti aplikácie, prípadne aj celkovú závislosť na tomto jazyku (Laddad, 2003).

Zaujímá nás preto uplatnenie jazyka pri umožnení konfigurovateľnosti konkrétnych vlastností v kóde, a samotné generovanie konkrétného variantu produktu s prípadnou požiadavkou nezávislosti implementácie od tohto použitého jazyka. Zhodnotíme prínosy a kvality aspektovo orientovaného prístupu na konkrétnej implementácii, a porovnáme ich s identifikovanými problémami a benefitmi spomenutými v rôznych reimplementáciách už existujúcich softvérových systémov ako je vedecká kalkulačka (Botterweck, 2009) alebo databázový systém (Kastner, 2007). Samotné analýzy naznačujú význam voľby domény pri zostrojovaní radu softvérových produktov, preto by sme v neposlednom rade chceli porovnať vplyv domény pri ich implementácii.

Plánujeme prispôsobiť existujúcu hru Battleship<sup>2</sup> tak, aby umožňovala manažovať vlastnosti na základe už vytvoreného modelu vlastností, väčšina ktorých nie je vôbec implementovaná. Cieľom je aj analyzovať spôsoby odvodenia konkrétného produktu, a mieru jeho závislostí od jazyka AspectJ.

Ďalšou aplikáciou pre analýzu je odvodzovanie rôznych fraktálov (ukážky typov nájdete v Pelánek, 2012) z pôvodného algoritmu, ktoré by sme realizovali zásahom aspektov do jeho vykonávania. Predpokladáme, že úloha samotného návrhu vlastností pre uvedenú aplikáciu je určená hodnotou vzhľadom fraktálu, pričom zhotovenie radu produktov preň môže vyžadovať iné nároky na model vlastností, ako napríklad generovanie všetkých možných derivácií pre

---

<sup>1</sup> <https://www.eclipse.org/aspectj/>

<sup>2</sup> <https://github.com/juletx/BattleshipFeatureIDE>

následné overenie estetiky a identifikovanie najvhodnejších kandidátov. Zamýšľame preto zhodnotiť význam prípadného potenciálu aspektovo orientovaného riešenia pre derivovanie konkrétnych fraktálov.

# Použitá literatura

BEUCHE, Danilo a Mark DALGARNO, 2006. Software Product Line Engineering with Feature Models. 2006, s. 7.

BOTTERWECK, Goetz, Kwanwoo LEE a Steffen THIEL, 2009. Automating Product Derivation in Software Product Line Engineering. 2009, s. 6.

KASTNER, Christian, Sven APEL a Don BATORY, 2007. A Case Study Implementing Features Using AspectJ. V: *11th International Software Product Line Conference (SPLC 2007): 11th International Software Product Line Conference (SPLC 2007)* [online]. Kyoto, Japan: IEEE, s. 223–232 [cit. 30.9.2021]. ISBN 978-0-7695-2888-5. Dostupné na: doi:10.1109/SPLINE.2007.12

LADDAD, Ramnivas, 2003. *AspectJ in action: practical aspect-oriented programming*. Greenwich, CT: Manning. ISBN 978-1-930110-93-9.

PELÁNEK, Radek, 2012. *Programátorská cvičebnice*. 1. vydání. Brno: Computer press. ISBN 978-80-251-3751-2.

YOUNG, Trevor J a B MATH, 1999. Using AspectJ to Build a Software Product Line for Mobile Devices. 1999, s. 73.