FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

Posudek oponenta závěrečné práce

Oponent práce:	Ing. Jiří Daněček
Student:	Patrik Hanes
Název práce:	Reaktivní technologie v jazyce Java
Obor / specializace:	Webové a softwarové inženýrství, zaměření Softwarové inženýrství
Vytvořeno dne:	6. února 2024

Hodnotící kritéria

1. Splnění zadání

- ▸ [1] **zadání splněno**
  [2] zadání splněno s menšími výhradami
  [3] zadání splněno s většími výhradami
  [4] zadání nesplněno

2. Písemná část práce 80 /100 (B)

Viz celkové hodnocení.

3. Nepísemná část, přílohy 85 /100 (B)

Viz celkové hodnocení.

4. Hodnocení výsledků, jejich využitelnost 75 /100 (C)

Celkové hodnocení 80 /100 (B)

V úvodu práce se student zmiňuje, „že na univerzitě se reaktivní programování moc do detailu neprobírá“. Zde bych rád upozornil, že je to jedno z témat předmětu BI-EJK, který je ovšem nepovinný.
Vzhledem k zaměření práce jako převážné rešeršní je zřejmé, že těžiště práce je v její teoretické části. V té se student pokusil osvětlit pojem reaktivního programování jako takového. To je samozřejmě obtížné, protože reaktivní programování je relativně nový přístup, který se stále vyvíjí a na jehož definici mohou být různé názory.
V teoretické část bohužel chybí několik zásadních informací, které jsou podstatné pro

pochopení významu reaktivního programování a jednotlivých etap, které k němu vedly. Tak např. při výkladu pojmu paralelismus opomněl zmínit, že rozeznáváme paralelismus skutečný realizovaný na úrovni hardwaru a paralelismus virtuální realizovaný přepínáním kontextu. Právě velká režie tohoto virtuálního paralelismu byla jedním z důvodů pro rozvoj reaktivního programování. To se snaží eliminovat blokování vláken, které je hlavní příčinou přepínání kontextu.
V sekci popisující populární reaktivní technologie v Javě jsou na obr 3.3 „What web frameworks do you use?" nelogicky smíchány skutečné webové technologie (JSF, Vadin, Spring MVC) s obecnými serverovými frameworky (Spring, Quarkus, Micronaut).
Při popisu specifikace Reactive Streams student opomněl zmínit, že využívají kombinace dvou základních způsobů komunikace paralelních agentů – aktivního konzumenta (pull-based) a aktivního producenta (push-based). Právě jejich kombinací dosahuje tato specifikace neblokované komunikace (která je inherentně přítomna v pull-based) a současně ochranu konzumenta před zahlcením (která je inherentně přítomno v pull based). Jelikož tato specifikace přes svůj název nedefinuje pojem streamu dat, vznikly na jejím základě abstraktnější knihovny, jako např. Mutiny používaná v Quarkus.
V práci dále není uvedeno, že reaktivní webové frameworky jsou speciálním případem reaktivních aplikaci, ve kterých se reaktivita dosahuje tak, že pokud možno co nejvíce requestů je zpracováno jedním tzv. IO vláknem. To ovšem vyžaduje pozorné programování, protože v handlerech requestů, které toto vlákno využívají, nesmí dojít k jeho blokaci.
Asi za největší nedostatek práce považuji, že se student nezmínil o dalším z možných řešení reaktivních systémů jako jsou korutiny v Kotlinu a virtuální vlákna v Javě.
V sekci Výkonnost jsem postrádal metriku propustnost, tedy jaké množství požadavků za vteřinu je daná technologie schopna zpracovat, aniž by se neúměrně prodloužila doba odezvy.
Co se týká naměřených hodnot, tak protože Quarkus používá knihovnu Mutiny, která je nadstavbou Vert.x, lze očekávat velkou korelaci mezi hodnotami naměřenými pro tyto dvě nikoliv nezávislé technologie.
Celkově práci hodnotím velmi dobře.

Instrukce

Splnění zadání

Posuďte, zda předložená ZP dostatečně a v souladu se zadáním obsahově vymezuje cíle, správně je formuluje a v dostatečné kvalitě naplňuje. V komentáři uveďte body zadání, které nebyly splněny, posuďte závažnost, dopady a případně i příčiny jednotlivých nedostatků. Pokud zadání svou náročností vybočuje ze standardů pro daný typ práce nebo student případně vypracoval ZP nad rámec zadání, popište, jak se to projevilo na požadované kvalitě splnění zadání a jakým způsobem toto ovlivnilo výsledné hodnocení.

Písemná část práce

Zhodnoťte přiměřenost rozsahu předložené ZP vzhledem k obsahu, tj. zda všechny části ZP jsou informačně bohaté a ZP neobsahuje zbytečné části. Dále posuďte, zda předložená ZP je po věcné stránce v pořádku, případně vyskytují-li se v práci věcné chyby nebo nepřesnosti.

Zhodnoťte dále logickou strukturu ZP, návaznosti jednotlivých kapitol a pochopitelnost textu pro čtenáře. Posuďte správnost používání formálních zápisů obsažených v práci. Posuďte typografickou a jazykovou stránku ZP, viz Směrnice děkana č. 52/2021, článek 3.

Posuďte, zda student využil a správně citoval relevantní zdroje. Ověřte, zda jsou všechny převzaté prvky řádně odlišeny od vlastních výsledků, zda nedošlo k porušení citační etiky a zda jsou bibliografické citace úplné a v souladu s citačními zvyklostmi a normami. Zhodnoťte, zda převzatý software a jiná autorská díla, byly v ZP použity v souladu s licenčními podmínkami.

Nepísemná část, přílohy

Dle charakteru práce se případně vyjádřete k nepísemné části ZP. Například: SW dílo – kvalita vytvořeného programu a vhodnost a přiměřenost technologií, které byly využité od vývoje až po nasazení. HW – funkční vzorek – použité technologie a nástroje, Výzkumná a experimentální práce – opakovatelnost experimentů.

Hodnocení výsledků, jejich využitelnost

Dle charakteru práce zhodnoťte možnosti nasazení výsledků práce v praxi nebo uveďte, zda výsledky ZP rozšiřují již publikované známé výsledky nebo přinášející zcela nové poznatky.

Celkové hodnocení

Shrňte stránky ZP, které nejvíce ovlivnily Vaše celkové hodnocení. Celkové hodnocení nemusí být aritmetickým průměrem či jinou hodnotou vypočtenou z hodnocení v předchozích jednotlivých kritériích. Obecně platí, že bezvadně splněné zadání je hodnoceno klasifikačním stupněm A.