UNIVERZITA PARDUBICE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

POSUDEK OPONENTA BAKALÁŘSKÉ PRÁCE

Jméno studenta: Bc. Jan Stehno

Název práce: Principy funkcionálního programování a jejich aplikace v moderních OOP jazycích

Autor posudku: Ing. Jan Merta, Ph.D.

1. Jaké metody (příslušející navazujícímu magisterskému studiu) diplomant ve své práci uplatnil?

Diplomant uplatnil pokročilé programátorské koncepty a hlubší znalost jazyka Java a jazyku Haskell, který není součástí studia.

2. Co diplomant při vypracování své DP vytvořil?

Diplomant vytvořil srovnání jazyka Java a jazyka Haskell z hlediska funcionálního programování. Popsal principy funkcionálního programování na příkladech a rozebral odlišné přístupy obou jazyků k tomuto paradigmatu a jednotlivé klady a zápory. Nakonec navrhl komplexnější příklad, který pro srovnání implementoval v jazyce Haskell a několika zpsůoby v jazyce Java (klasický přístup, funcionální pomocí streamů, využití knihoven Vavr apod., které jsou podobnější Haskellu). Autor rozebírá i problematiku property-based testing (testování obecných vlastností funkcí) v Javě pomocí Junit-Quickcheck a Jqwik.

3. Jak diplomant prokázal správnost navrženého řešení problému?

Diplomant otestoval funkčnost všech příkladů.

4. Podařilo se diplomantovi splnit cíle práce, které mu byly uloženy?

Ano.

5. Jaká je kvalita textu diplomové práce z hlediska jeho struktury, srozumitelnosti, jazykové a typografické úrovně?

Text práce je celkově kvalitní s logickou strukturou, srozumitelný bez výrazných jazykových a typografických prohřešků. Práce s literaturou má dobrou úroveň, autor čerpal zejména z odborných knižních i online zdrojů.

6. Jak byla vyhodnocena kontrola textu DP (případně zdrojových kódů softwaru) pomocí systému pro odhalování plagiátů mezi závěrečnými pracemi?

Práce má podle systému pro kontrolu plagiátů nejvyšší shodu 1 %, nepovažuji proto za plagiát.

7. Které nejasnosti vyskytující se v DP by měl diplomant objasnit při obhajobě a jaké jsou další připomínky k DP?

Připomínky:

- strana 20: Autor v kapitole 1.5 zmiňuje o architekturách „bez nutnosti serveru“, jde nejspíše o překlad architektur serverless. Toto může psůobit dojmem, že se v této architektuře nepoužívají servery. Serverless model cloudového provedení, kde poskytovatel cloudu dynamicky spravuje zřizování, škálování a údržbu serverové infrastruktury. Je však nutné podotknout, že zavádějící je už anglický originál.

- strana 25: Příklady s incializací číselných proměnných by mohly být lépe okomentovány, aby bylo jasné, že si Haskell umí i typy odvodit.
- strana 34: příklad kódu funkce pro rozlišení sudého/lichého čísla mi přijde těžkopádný (využití where s *c*)
    - calculateOddEven :: Int -> String
      calculateOddEven x
      | c == 0 = "Výsledek je 0"
      | even c = "Výsledek je sudý"
      | odd c = "Výsledek je lichý"
      where c = x * 2 – 3
    - proč se tam musí počítat c = x * 2 – 3?
- strana 17: „Taktéž je možné libovolně paralelizovat a prohazovat pořadí výpočtů, protože závislosti tvoří strom.“
    - Libovolná paralelizace je samozřejmě možná jen do jisté míry, pokud výsledek funkce A například závisí na výsledku jiné funkce B, není možné počítat B před A. Toto autor později v práci zmiňuje. Ve stromu lze paralelizovat sousední podstromy ale ne otce před potomkem.
    - Jsou závislosti mezi funkcemi vždy opravdu jenom strom?

8. V závěru je nutno jednoznačně uvést, zda je práce doporučena či nedoporučena k obhajobě a jakým klasifikačním stupněm je hodnocena?

Celkově jde o kvalitní diplomovou práci. Práci doporučuji k obhajobě.

Navržená výsledná známka: A

V Pardubicích**, dne 2. září 2025**

_____

podpis