

## OPONENTSKÝ POSUDEK BAKALÁŘSKÉ PRÁCE

Student: **Tomáš Bartošík**

Oponent: **Ing. Tomáš Juřena**

Studijní program: **Inženýrská informatika**

Studijní obor: **Softwarové inženýrství**

Akademický rok: **2019/2020**

Téma bakalářské práce: **Inovace úloh v kurzu jazyka C++**

### Hodnocení práce:

	A	B	C	D	E	F
	Hodnocení: A – nejlepší; F - nevyhovující					
1. Aktuálnost řešeného tématu	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2. Obtížnost zadaného úkolu	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3. Splnění všech bodů zadání	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4. Vhodnost zvolené metody řešení	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5. Logické členění práce	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6. Úroveň jazykového zpracování	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7. Formální úroveň práce	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8. Práce s literaturou a její citace	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9. Úroveň zpracování teoretické části	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10. Kvalita zpracování praktické části	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11. Dosažené výsledky práce	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12. Přínos práce a její využití	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

### Celkové hodnocení práce:

Výsledná známka není průměrem výše uvedených hodnocení. Znamku uvede oponent dle svého uvážení dle klasifikační stupnice ECTS:

A – výborně, B – velmi dobře, C – dobře, D – uspokojivě, E – dostatečně, F – nedostatečně.

Stupeň F znamená též „nedoporučuji práci k obhajobě“.

**Předloženou bakalářskou práci doporučuji k obhajobě a navrhuji hodnocení  
C - dobře.**

V případě hodnocení stupněm „F – nedostatečně“ uveďte do připomínek a slovního vyjádření hlavní nedostatky práce a důvody tohoto hodnocení.

### Otázky k obhajobě:

Proč nebyla využita novější specifikace jazyka C++ než c++11?

Uvažoval jste o testování v Docker kontejnerech pro zaručení rovných podmínek pro testování?

Proč jste nevygeneroval dokumentaci z anotací Doxygen a nepřiložil ji k zadáním v čitelné podobě?

Na základě jakých požadavků byla vybrána témata pro úkoly?

### Další připomínky, vyjádření, náměty k obhajobě práce (možno pokračovat i na další stránce):

Autor předkládané práce se v teoretické části zabývá syntaxí jazyka C++ a rozdíly proti jazyku C, dále je rozebrána problematika objektově orientovaného programování, a jsou detailněji popsány principy objektového návrhu v jazyce C++. Následně je čtenář seznámen s problematikou testování,

konvencích při tvorbě kódu a programové dokumentace, a o různých možnostech automatické kontroly. V praktické části jsou velmi detailně diskutovány nově vytvořené úlohy, návrh systému pro automatickou kontrolu s využitím software Gitlab, a je provedena analýza řešení jedné úlohy studenty kurzu.

Práce je na tento typ práce velmi rozsáhlá a detailní. Nevyskytuje se v ní velké množství gramatických chyb, ale celkově není příliš čtivá. Toto je způsobeno komplikovanými souvětími, která místy postrádají smysl nebo návaznost na předcházející text. Práce obsahuje 37 literárních zdrojů, ze kterých je 6 citací knih. Většina použitých citací je dle doporučení pro jednotnou úpravu kvalifikačních prací, pouze některé tomuto doporučení neodpovídají.

Dále jsou v práci zaměňovány technické termíny (např. nástroj-knihovna, program-modul nebo instance-ukazatel) a obsahuje i faktické chyby. Například tvrzení v kapitole 1.5.3 „Tím pádem ukazatele na objekt typu *shared\_ptr* sdílí vlastnictví objektu, tento objekt je z paměti uvolněn až při odstranění posledního z existujících *shared\_ptr* ukazatelů.“, kde autor zaměňuje instanci objektu *shared\_ptr* za ukazatel na tuto instanci, nebo tvrzení z kapitoly 4.3 „... - neúspěch jakékoliv fáze zabraňuje dalšímu vykonávání kontroly.“, které není pravdivé.

Vytvořené úlohy jsou nejprve zaměřeny na syntaxi a objektový vývoj v jazyce C++, ale pozdější úkoly jsou spíše algoritmické a studenti procvičují nabyté znalosti. V úkolech není řešena například problematika referencí, přesunová sémantika, scoped výčet, parameter-pack a jeho expanze, datový typ *auto*, *range-loops* a další, což bych při použití standardu c++11 očekával. V úkolech je patrné, že jsou tvořeny spíše v jazyce C a částečným použitím objektového přístupu. Jedním z indikátorů tohoto problému je to, že nejsou používány funkce z *std* namespace, ale jejich C varianty.

Vypracování úkolů je konzistentní a dodržuje definovanou strukturu a stanovená pravidla. Jedinou výjimkou je úkol „Dědičnost, virtuální funkce a polymorfismus“, u kterého nejsou dodrženy názvy funkcí (chybí prefix názvu třídy) jako ve zbytku úloh. Použití prefixu *mi* v zadáních vadí vzhledem k výsledné délce názvu funkce v implementačním souboru a limitu osmdesáti znaků na řádek.

Prefix je dle mého názoru navíc zbytečný a znepřehledňuje kód. U tohoto úkolu se navíc projevila nevhodnost tohoto pojmenování, protože při dědění společné funkcionality je nevhodné mít členskou funkci pojmenovanou s prefixem, jelikož tento název se přenáší dále do potomků.

Použití Doxygen anotací vítám, ale pro studenta bez vývojového prostředí podporující zobrazení těchto anotací, je méně přívětivé se v dokumentaci zorientovat. Pokud by byla vygenerovaná dokumentace dostupná v repozitáři, tak by to mohlo být pro studenty přívětivější.

I přes zmíněné nedostatky práce splňuje požadavky na bakalářskou práci a doporučuji ji k obhajobě.