

PRAXE

Číslo úlohy	STŘEDNÍ PRŮMYSLOVÁ ŠKOLA CHOMUTOV	Jméno
Datum zadání	NÁZEV ÚLOHY	Třída

Zadání

Úplné zadání řešené úlohy se seznamem všech požadavků. Postačuje ve stručné formě např. bodového seznamu.

Teorie

Teoretické znalosti potřebné k řešení. Popis principů a postupů nutných k vyřešení úlohy spolu s popisem použitých technologií a práce s nimi. V kostce „*co musím znát abych vůbec mohl začít řešit*“.

Popis programu

Popis zrealizovaného řešení s vazbou na vlastní kód programu, upozornění na zajímavé části programu a místa, kde jsou řešeny klíčové funkcionality. Popis programátorských technik použitých v programu.

Jedná se o popis vnitřního chování a vazeb v program ve stylu: „*Po načtení kódu se automaticky aktivuje rutina XYZ, které v závislosti na stavu vlastnosti ABC třídy provede automatické vyhodnocení vstupních dat a buď je předá k uložení voláním metody OPQ na instanci třídy ZpracujData, nebo zobrazí chybové hlášení popisující problém*“.

V případě dokumentace reálného projektu by tato část sloužila jako popis celkové architektury řešení.

Pozor! Stejně jako v ostatních bodech, vytváříte *programátorskou (technickou)* dokumentaci, nikoliv uživatelskou. Generálně jakýkoliv popis programu, který sklouzne to formy uživatelského manuálu („*po stisknutí tlačítka je možné v druhém formuláři zadat potřebné údaje a potvrdit tlačítkem OK*“) bez jakéhokoliv popisu interního chování aplikace **je špatně**.

Rozbor proměnných a funkcí (metod)

Seznam použitých proměnných (pouze proměnné důležité pro hlavní část algoritmu), jejich typů a použití v programu.

Seznam použitých metod (funkcí) důležitých pro hlavní část algoritmu, jejich parametrů a operace, kterou provádějí.

Důležitá je čitelnost a přehlednost této sekce, která v reálné dokumentaci plní funkci rychlé reference a v případě důsledně psaného kódu je možné ji automaticky generovat různými nástroji (*Doxygen* apod.). Ideální formou je tabulkový zápis, případně seskupený po jednotlivých (hlavních) třídách, případně souborech použitých v kódu.

Vývojový diagram

Grafický popis **klíčových** částí algoritmu. Nemusí obsahovat popis GUI programu. Mimo jádro úlohy je možné používat zjednodušování, případně odkazovat komentářové bloky pro bližší vysvětlení významu.

Nemusí se jednat pouze o vývojový diagram (ikdyž je nejjednodušší grafickou formou vyjádření algoritmu), můžete použít i např. UML nástroje, diagram komunikací, stavové a sekvenční diagramy s přechodovými podmínkami apod. Vždy však **úplně a správně**.

Opět je důležitá čitelnost a přehlednost, vývojový diagram by v žádném případě neměl být přepisem existujícího kódu (dle definice je dokonce na použitém jazyku nezávislý). Na první pohled musí být jasný princip algoritmu, který byl (měl by být) realizován v plném rozsahu (bez ohledu na míru splnění v rámci cvičení). Pro rychlou kontrolu správnosti lze použít kritérium „*zvládl bych podle takto popsáního algoritmu pochopit co mám programovat a sestavit funkční kód?*“

Schéma zapojení

Schéma zapojení umožňuje správné připojení periferie k řídicímu systému. Jedná se o **skutečné elektrické schéma** s úrovní detailu potřebnou pro řešení úlohy a zachycení potřebných ovládacích signálů. V některých případech se může jednat o schéma převzaté z dokumentace úlohy (s kvalitou srovnatelnou s ostatními částmi odevzdané dokumentace) – např. pro složitější periferie, kde se schéma hodí i pro vysvětlení

principů ovládání v rámci části teorie.

Opět by se mělo jednat o přehlednou rychlou referenci, v ideálním případě vytvořenou v rámci prvotního seznámení s úlohou. Během psaní kódu (minimálně rutin pro ovládání technologií) je často potřeba informace typu „*kde je umístěn signál XYZ, který bit výstupu ovládá signál PQR*“, kde okamžitou odpověď by mělo dát schéma zapojení. **Fotografie či malůvka propojení je v tomto ohledu informačně nedostačující.**

Komentovaný výpis programu

Komentovaný výpis klíčových částí programu. Dokumentace je odevzdávána v elektronické podobě, takže může obsahovat kompletní výpis kódu (bez nepotřebných servisních či konverzních rutin, kde stačí pouze hlavičky funkcí s parametry bez kódu). V případě, kdy není řešení kompletní, je možné vložit i např. naznačení dalšího rozšíření kódu (minimálně přeloženého kompilátorem bez syntaktických chyb).

Pro výpisy používejte neproporcionální font (*Courier, Terminal* apod.), pro zvýšení čitelnosti a přehlednosti kódu jej vkládejte se zvýrazněním syntaxe (např. kopie se zachováním formátu z *PS Padu* nebo některé z online služeb (např. <http://hilit.me>)

Správný zdrojový kód by měl být přehledný (viditelnost nadřazenosti a podřazenosti bloků, začátku a konce smyček, vnitřního kódu podmínek) a *komentovaný*. Komentáře by měly vznikat během psaní kódu a obsahovat důvody a návaznosti na další zpracování, nikoliv „*komentovat příkaz*“ - srovnajte úryvky níže:

```
int state = PS_RUNNING; // do proměnné stav hodnotu konstanty PS_RUNNING
int state = PS_RUNNING; // aktualizace příznaku pro další zpracování paketu
```

Pozor na čitelnost vloženého kódu, použijte barevné schéma vhodné i pro případný tisk dokumentace. Tmavé pozadí s titěrným fontem není příjemné ani pro elektronické čtení, natožpak pro tisk.

Pokud je kód složený z několika dílčích souborů (měl by být), je nutná také přehledná identifikace o jaký soubor se jedná (viz. např. spojení s přehledem proměnných a funkcí).

Odpovědi na otázky

Několika větami zodpovězené otázky doplňující a rozšiřující řešenou úlohu (viz. samostatný dokument), které by vás měly „*donutit*“ zamyslet se nad souvislostmi, na které není v rámci cvičení prostor. Odpovídejte vlastními slovy (nejlepší způsob jak si ověřit pochopení problému je vlastní formulace vysvětlení), copy & paste vložení informací z externích zdrojů popírá smysl tohoto bodu a není ničím přínosné (ve čtvrtém ročníku jistě text kopírovat umíte).

Nezapomeňte na korektní uvedení zdrojů, nejlépe pomocí číslovaných poznámek pod čarou (bibliografická citace a seznam literatury není požadován). Anglická Wikipedie není špatná pro získání prvotního náhledu na problematiku (o české verzi to povětšinou neplatí). Jako při všech pracích s informacemi byste měli mít dané tvrzení ověřeno z několika nezávislých (a dostatečně validních) zdrojů. Pokud po přečtení vysvětlení z použitého zdroje text nechápete, najděte si jiný s lepším vysvětlením. *Použití informace odkudkoliv bez uvedení zdroje je špatně.*

Závěr

Zhodnocení úlohy, možnosti jejího rozšíření, úprav či použití pro další aplikace. Popis slabých míst realizovaného algoritmu, zdůvodnění použitého řešení, návrh možných změn a úprav algoritmu, případně paměťová a časová náročnost algoritmu.

Tato část je na rozdíl od všech ostatních vašim subjektivním shrnutím úlohy, zhodnocením co se povedlo, nepovedlo, kde jste se na řešení zadrhli, co byste se současnými znalostmi úskalí úlohy dělali jinak včetně např. jiného rozložení kódu, na které během cvičení nezbýval čas.

Závěry stylu „*Úloha se mi moc líbila, nebyla moc těžká a už se těším na další*“ **NEJSOU ZÁVĚREM ÚLOHY !!!**

Dokumentaci odevzdáváte v podobě PDF souboru **se záložkami pro rychlou navigaci** spolu se zdrojovým kódem (a **jenom zdrojovým kódem**) aplikace v **zip** archivu pomocí emailu.