

Techniki Kompilacji – projekt

Zalecenia ogólne

Proponowane problemy projektowe są zasadniczo przeznaczone do opracowania *indywidualnego*. Mogą sporadycznie pojawiać się propozycje projektów zespołowych, ale wtedy proponowany będzie podział odpowiedzialności pomiędzy członków zespołu. Opracowywane programy powinny spełniać następujące ogólne kryteria:

Język programowania. Projekty powinny być opracowane w standardowym C++. Z biblioteki standardowej można korzystać bez ograniczeń; trzeba natomiast zaniechać używania rozszerzeń dostępnych tylko lokalnie. Program powinien być przenośny i dać się skompilować/wykonać zarówno na platformie Linux (g++) jak i Windows (np. Visual C++ 2008, 2010). Korzystanie z innych języków implementacji jest możliwe po uzgodnieniu z opiekunem projektu.

Samodokumentowanie. Każdy projekt powinien zawierać plik tekstowy **readme.txt**, w którym należy umieścić zwięzłe informacje dotyczące przeznaczenia i funkcji opracowanego projektu, organizacji kodu źródłowego oraz podstawowe dane o korzystaniu z programu. W szczególności powinny tam znaleźć się:

- "wizytówka" studenta
- tytuł i krótka specyfikację problemu
- specyfikacja aktywacji programu, stosowanych parametrów i opcji
- opis konwencji dotyczących danych wejściowych i prezentacji wyników
- informacje o funkcjonalnej dekompozycji programu na moduły źródłowe - nagłówkowe i implementacyjne ("przewodnik" po plikach źródłowych)
- inne informacje dodatkowe o szczególnych decyzjach projektowych (np. ograniczenia, sposób reagowania na błędy, obsługa sytuacji wyjątkowych, itp.).

Każdy plik źródłowy musi być opatrzony komentarzem nagłówkowym z nazwiskiem studenta i tytułem problemu.

Samoprezentacja programu. Program powinien zachowywać się bez tajemnic użytkowych. W szczególności jeden z typowych sposobów aktywacji programu (np. wg konwencji systemu UNIX) powinien pokazywać wszystkie prawidłowe formy aktywacji lub kierować do pliku z informacjami dodatkowymi (np. **readme.txt**).

Wybór konwencji we-wy. Programy opracowywane są zasadniczo dla we-wy znakowego. Postać danych i wyników powinna być określona ściśle i czytelnie, zwykle przy pomocy odpowiedniej gramatyki. W pewnych przypadkach korzystne jest posłużenie się *schematem translacji* jako standardowym opisem realizowanego przetwarzania.

Styl. W programie należy przestrzegać zasady dobrego stylu kodowania (przejrzysty układ tekstu programu uwzględniający struktury zagnieżdżone; odpowiednia zawartość i porządek konstrukcji językowych w plikach nagłówkowych i plikach implementacyjnych; stosowanie identyfikatorów ułatwiających zrozumienie kodu; zwięzłe komentarze dla funkcji nietrywialnych; komentarze objaśniające wewnątrz funkcji). Nazwy plików powinny sugerować ich zawartość i rolę w programie. Projekt można wykorzystać także do przeciwiczenia dokumentowania kodu przy pomocy wybranego narzędzia (np. **Doxygen**).

Dokumentacja. Wraz z programem źródłowym trzeba przekazać zwięzłą dokumentację końcową w formie elektronicznej (dokument Word, OpenOffice lub pdf) zawierającą opis problemu i rozwiązania, opis wykorzystywanych struktur danych i algorytmów pomocniczych. Przy opracowywaniu dokumentacji proszę kierować się prostą wskazówką: *opis ma być taki, aby zadowolili mnie samego gdybym o problemie i jego rozwiązaniu miał dowiedzieć się z tego dokumentu*. Ewentualne dodatkowe ustalenia szczegółowe dotyczące zawartości dokumentacji końcowej należy uzgodnić z opiekunem projektu.

Terminy. Do **7 grudnia 2012** trzeba przygotować koncepcję rozwiązania problemu i przedstawić ją prowadzącemu projekt do zaakceptowania. Termin zakończenia projektu (prezentacja ostatecznej wersji rozwiązania i przekazanie dokumentacji końcowej) przypada na dzień **18 stycznia 2013**.

Niedotrzymanie terminu przedstawienia projektu wstępnego może spowodować obniżenie oceny o 5p za tydzień zwłoki.

Konsultacje. Terminy konsultacji związanych z projektem zostaną zaproponowane przez opiekunów projektów. Zawsze istnieje możliwość uzyskania dodatkowych konsultacji w terminach uzgodnionych z prowadzącym.