

PROE – temat projektu nr 1 dla grupy wtorkowej g. 10-12

Zadanie

Zadanie polega na zaprojektowaniu i zakodowaniu w C++ modelu dowolnego „systemu”, składającego się z obiektów należących przynajmniej do trzech klas, nazwanych tutaj dla ustalenia uwagi *Elementem*, *Kolekcją* i *Aplikacją*. Nazwy te mają oczywiście charakter roboczy – w rzeczywistym projekcie powinny one lepiej odzwierciedlać specyfikę modelowanej dziedziny, do której należy system. Załóżmy, że jest to firma transportowa z pojazdami kilku różnych typów (przykład zastrzeżony na potrzeby tej instrukcji – proszę wymyślić własny). *Elementami* są tu więc pojazdy. Powinny one charakteryzować się zarówno pewnymi cechami wspólnymi, jak i odrębnymi. Na przykład każdy pojazd ma określony zasięg, lecz autokar jest dodatkowo charakteryzowany przez maksymalną liczbę pasażerów, zaś ciężarówka przez ładowność.

Działanie programu

Program rozpoczyna działanie od pobrania informacji o poszczególnych *Elementach*. Mogą one zostać wczytane z pliku lub z wejścia standardowego. Na tej podstawie kolejne obiekty reprezentujące *Elementy* różnych typów są tworzone i dodawane do wspólnej *Kolekcji*, posiadającej m.in. dynamicznie alokowane tablice *Elementów* – po jednej dla każdego typu. Przy wyjściu z programu aktualny stan *Kolekcji* jest zapisywany w pliku, który przy ponownym uruchomieniu może posłużyć do przywrócenia jego stanu sprzed zamknięcia.

Uwagi implementacyjne

Kod programu powinien być podzielony na pliki .cpp zawierające implementację klas i .h zawierające ich deklaracje.

Aplikacja jest jedynym składnikiem systemu, który komunikuje się ze światem zewnętrznym, np. poprzez pliki i/lub wejście czy wyjście standardowe. Stanowi ona więc warstwę pośredniczącą między użytkownikiem a resztą programu. W programie musi istnieć dokładnie jeden egzemplarz klasy *Aplikacja*.

Ponieważ informacja o liczbie *Elementów* nie jest dostępna w momencie kompilacji programu, struktura danych w klasie *Kolekcja* musi mieć możliwość dopasowywania swoich wymiarów w miarę dodawania do niej kolejnych *Elementów* (niekoniecznie „na styk” – warto zapewnić pewien zapas, by uniknąć zbyt częstej realokacji pamięci i kopiowania zawartości). UWAGA! Taką elastyczną strukturę danych należy zaprojektować samodzielnie. Nie korzystamy z gotowych kontenerów typu `std::vector` itp.

Wszystkie klasy, w których wykorzystywana jest dynamiczna alokacja pamięci, powinny mieć zdefiniowany odpowiedni destruktor.

Należy korzystać ze wszystkich dobrodziejstw języka C++ przedstawionych na wykładzie:

- do zarządzania pamięcią wykorzystujemy polecenia *new/delete* zamiast *malloc/free*,
- do wymiany danych z terminalem i plikami używamy strumieni, a nie poleceń *printf, scanf* itp.,
- ograniczamy do minimum liczbę funkcji niebędących metodami klas.

Sprawozdanie

Wynik prac proszę przesłać na adres prowadzącego: **dkasprow@imio.pw.edu.pl** Wyniki powinny obejmować następujące elementy:

1. Sprawozdanie **w formacie PDF**, a w nim:
 - Krótki opis zadania, jakie realizuje program.
 - Diagram klas w konwencji UML.
 - Krótki opis klas i najważniejszych atrybutów i metod. Ten punkt można pominąć, o ile wygeneruje się odpowiednią dokumentację, np. za pomocą programu *Doxygen*.
 - Opis sposobu użycia programu (format koniecznych plików wejściowych, sposób wprowadzania danych z klawiatury itp.) a także sposób kompilacji, o ile nie jest on trywialny.
 - Ewentualne zrzuty ekranu pokazujące przebieg działania, np. komunikaty generowane przez program.
2. Projekt skompresowany do formatu *.zip/.tar.gz*. Powinien on zawierać:
 - Pliki z kodem i pliki nagłówkowe.
 - Plik/pliki umożliwiające kompilację – *makefile* lub struktura plików i katalogów projektu środowiska CodeBlocks lub Microsoft Visual Studio. **Program musi dać się skompilować i uruchomić na komputerach stanowiących wyposażenie pracowni, w której odbywają się zajęcia.**
 - Ewentualne pliki wejściowe, zawierające dane, parametry konfiguracyjne itp.
3. Jeśli wygenerowano dokumentację za pomocą programu *Doxygen*, proszę o jej dołączenie do archiwum wspomnianego w powyższym punkcie.