PROE – tematy projektu nr 1
dla grup piątkowych 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 różnymi pojazdami (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 klasy *Element* są tworzone i dodawane do wspólnej *Kolekcji*, posiadającej m.in. dynamicznie alokowaną tablicę *Elementów*.

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*. Z tego względu powinna ona implementować wzorzec projektowy *Singleton*.

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.

Podany przykład z pojazdami bardzo różnych typów, charakteryzującymi się jednak pewnymi cechami wspólnymi, sugeruje oczywiście użycie mechanizmu dziedziczenia. *Autokar* i *Ciężarówka* będą więc dziedziczyły po klasie *Pojazd*. Samo dziedziczenie nie rozwiązuje do końca problemów związanych zarówno z przechowywaniem tych obiektów we wspólnej *Kolekcji*, jak i ich obrazowania w pliku graficznym. Oba te zadania można zrealizować w sposób dużo prostszy i bardziej elegancki dzięki zastosowaniu polimorfizmu. Proszę pamiętać, że programowanie obiektowe i charakterystyczne dla niego mechanizmy powstały po to, by uprościć projektowanie i kodowanie aplikacji!

Sprawozdanie

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

1. Sprawozdanie **w formacie PDF**, a w nim:
    1. Krótki opis zadania, jakie realizuje program.
    2. Diagram klas w konwencji UML.
    3. Krótki opis klas i najważniejszych atrybutów i metod. Ten punkt można pominąć, o ile wygeneruje się odpowiednią dokumentację, np. za pomocą programu *Doxygen*.
    4. Opis sposobu użycia programu (format koniecznych plików wejściowych, sposób wprowadzania danych z klawiatury itp.) a także sposób kompilacji, o ile nie jest on trywialny.
    5. Ewentualne zrzuty ekranu pokazujące przebieg działania, np. komunikaty generowane przez program.
2. Kod programu wraz z niezbędnymi plikami wejściowymi (zawierającymi dane, parametry konfiguracyjne itp.). **Program musi dać się skompilować i uruchomić na komputerach stanowiących wyposażenie pracowni, w której odbywają się zajęcia.** Kod proszę spakować do archiwum *.zip* lub *tar.gz*.
3. Jeśli wygenerowano dokumentację za pomocą programu *Doxygen*, proszę o jej dołączenie do archiwum wspomnianego w powyższym punkcie.

Wszelkie elementy tekstowe sprawozdania (opis projektu, plik wejściowy, kod itp.) powinny być zapisane w pliku PDF jako tekst – niedopuszczalne jest wklejanie tych elementów jako obrazów.

Dominik Kasprowicz i Krzysztof Siwiec, 22.03.2018