PROE – tematy projektu nr 2
dla grup piątkowych g. 10-12

Zadanie

Zadanie polega na rozszerzeniu funkcjonalności zadania z pierwszego projektu o następujące elementy:

- Użycie wyjątków do obsługi sytuacji nietypowych, np. brak pliku wejściowego, brak możliwości zapisu pliku wynikowego itp. Własne klasy wyjątków powinny dziedziczyć po *std::exception* i implementować zadeklarowaną tam czysto wirtualną funkcję *what*().
- Wykorzystanie wybranego kontenera z biblioteki STL (*Standard Template Library*) do implementacji wspólnej kolekcji wskaźników na obiekty wszystkich klas wyprowadzonych z klasy *Element* (według terminologii użytej w instrukcji do Projektu 1). **UWAGA** Klasa *Kolekcja* nie powinna dziedziczyć po kontenerze z biblioteki STL, lecz taki kontener zawierać. Powinna udostępniać własny interfejs, pozwalający na bezpieczne dodawanie i usuwanie *Elementów*.
- Polimorfizm. Klasa podstawowa *Element* musi być abstrakcyjna.
- Dziedziczenie wielopoziomowe – przynajmniej jedna z klas wyprowadzonych z klasy *Element* powinna mieć klasy potomne.
- Wielodziedziczenie – przynajmniej jedna z klas wyprowadzonych powinna dziedziczyć po więcej niż jednej klasie. W przykładzie z bazą pojazdów moglibyśmy mieć np. klasę *Pojazd*, z której wyprowadzone zostałyby klasy (konkretne, tj. nie abstrakcyjne!) *Spychacz* i *Koparka*. Z kolei po tych dwóch klasach dziedziczyłaby klasa *Koparkospycharka*.

Podczas tworzenia projektu należy mieć również na względzie następujące wytyczne:

- Unikamy niepotrzebnych kopiowań i konstruowania obiektów. Gdziekolwiek jest to możliwe i sensowne, obiekty przekazujemy przez wskaźnik lub referencję.
- Odpowiednio zarządzamy pamięcią. Gdziekolwiek jest to możliwe i sensowne, obiekty tworzymy dynamicznie, przydzielając im pamięć za pomocą operatora *new*. Jeżeli wśród składowych klasy znajdują się wskaźniki do obiektów tworzonych dynamicznie, klasa ta musi mieć odpowiednio zdefiniowany destruktor.
- Każda klasa – dla ustalenia uwagi nazwijmy ją *Moja* – jest zadeklarowana (a w przypadku

klas szablonowych – także częściowo zaimplementowana) w osobnym pliku nagłówkowym *Moja.h* i zaimplementowana w pliku *Moja.cpp* (lub *Moja.hpp*, jeśli jest szablonem klasy).

- Gdziekolwiek jest to możliwe i sensowne, korzystamy z dobrodziejstw języka C++, np. klasy *std::string* zamiast tablic znaków, operatora *new* do alokacji pamięci, kolekcji z biblioteki STL (kolejek, wektorów, list itp.) zamiast tablic, iteratorów zamiast indeksów itp.
- Mile widziane jest stosowanie wskaźników „inteligentnych”: *unique_ptr* i/lub *shared_ptr*.

Wynik prac

Wynikiem projektu powinny być następujące elementy:

1. Kod programu – katalog spakowany do pliku ZIP lub TAR.GZ, koniecznie z Makefile pozwalającym na jego kompilację i zlinkowanie za pomocą programu *Make* i kompilatora *g++* lub *clang*. Alternatywą jest użycie do tego celu programu *SCons*.
2. Sprawozdanie w formacie PDF. Powinno zawierać te same elementy, co w Projekcie 1 (oprócz kodu, który przesyłamy w spakowanym katalogu), a więc:
   a) Opis struktury i funkcjonowania projektu w języku naturalnym.
   b) Diagram klas w formacie UML.
   c) Wyniki testowania, a w szczególności:
      - Zawartość pliku wejściowego (jeśli program pobiera dane z pliku) lub sekwencję danych wprowadzanych z klawiatury.
      - Ewentualnie zrzuty ekranu pokazujące przebieg działania, np. komunikaty generowane przez program.

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

Dominik Kasprowicz i Krzysztof Siwiec, 19.04.2018