PROE – tematy projektu nr 2
dla grupy wtorkowej g. 10-12

Zadanie

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

- Polimorfizm dynamiczny. 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 mogłaby istnieć np. klasa *Pojazd*, z której wyprowadzone zostałyby klasy (konkretne, tj. nie abstrakcyjne!) *Spychacz* i *Koparka*. Z kolei po tych dwóch klasach dziedziczyłaby klasa *Koparkospycharka*.
- Wykorzystanie przynajmniej jednego kontenera z biblioteki standardowej. Kontener taki powinien zostać użyty m.in. do implementacji wspólnej kolekcji wskaźników na obiekty wszystkich klas wyprowadzonych z klasy *Element* (według terminologii użytej w instrukcji do Projektu 1). Staramy się przy tym korzystać z iteratorów, zamiast z indeksów.
- Użycie wyjątków do obsługi sytuacji nietypowych, np. brak pliku wejściowego, brak możliwości zapisu pliku wynikowego itp. Oprócz wykorzystania standardowych klas wyjątków (wszędzie tam, gdzie jest to konieczne) należy stworzyć i wykorzystać przynajmniej jedną własną klasę wyjątku, dziedziczącą po *std::exception* i implementującą zadeklarowaną tam czysto wirtualną funkcję *what*().

Podczas tworzenia projektu należy mieć również na względzie następujące wytyczne:

- Staramy się używać „sprytnych” wskaźników: *unique_ptr* lub *shared_ptr* (w zależności od potrzeby).
- Unikamy niepotrzebnych kopiowań i konstruowania obiektów. Gdziekolwiek jest to możliwe i sensowne, obiekty przekazujemy przez wskaźnik lub referencję, ewentualnie stosujemy semantykę przenoszenia.
- Odpowiednio zarządzamy pamięcią. 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 w osobnym pliku nagłówkowym *Moja.hpp* i zaimplementowana w pliku *Moja.cpp* (lub *Moja.tpp*, jeśli jest

szablonem klasy).

- Gdziekolwiek jest to możliwe i sensowne, korzystamy z dobrodziejstw języka C++, np. klasy *std::string* zamiast tablic znaków itp.

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++*. **Program musi dać się skompilować i uruchomić na systemie Linux na komputerach stanowiących wyposażenie pracowni, w której odbywają się zajęcia.**
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, 20.11.2018