

[PROI] Projekt 4.

Adrian Brodzik

28 maja 2019

Zadanie

Zaprojektować prosty edytor graficzny, w którym można rysować obiekty proste (np. kółko) oraz złożone (zbiór obiektów). Obrazy generować w formacie *Scalable Vector Graphics (SVG)*. Zachować odpowiedni stopień abstrakcji i możliwość rozwoju aplikacji, w tym dodawanie kolejnych formatów plików.

Rozwiązanie

Niech `Attribute` będzie prostą strukturą danych postaci (`klucz`, `wartość`). Niech `Object` będzie zbiorem danych typu `Attribute`. Wtedy `Object` może reprezentować figurę o pewnych cechach typu `Attribute`. Taka separacja klas pozwala figurom posiadać dowolną liczbę cech oraz ułatwia generowanie plików zawierających te figury.

Niech `Double` będzie prostą strukturą danych zawierającą liczbę rzeczywistą oraz wartość logiczną. Niech `AttributeDouble` dziedziczy `Attribute` i przechowywał wartości typu `Double`. Wtedy możemy przechowywać cechy będącymi współrzędnymi absolutnymi lub relatywnymi (tzn. skalowanymi).

Niech `Complex` dziedziczy `Object` i będzie zbiorem danych typu `Object`. Wtedy `Complex` pozwala tworzyć złożone figury i obrazki.

Testowanie

Testy przeprowadzono za pomocą biblioteki `doctest` i serwisu ciągłej integracji `Travis CI`. Testowano konstruowanie figur prostych i złożonych, dodawanie i modyfikacja atrybutów oraz generowanie obiektów o relatywnych wymiarach i współrzędnych.

Przykład użycia

```
#include "simplegraphics/simplegraphics.hpp"

namespace SG = SimpleGraphics;

int main()
{
    SG::Complex img;

    img.add(SG::Rectangle(0, 0, 100, 50, "#FF0000"));
    img.add(SG::Rectangle(50, 50, 50, "#00FF00"));
    img.add(SG::Circle(25, 75, 25, "#0000FF"));

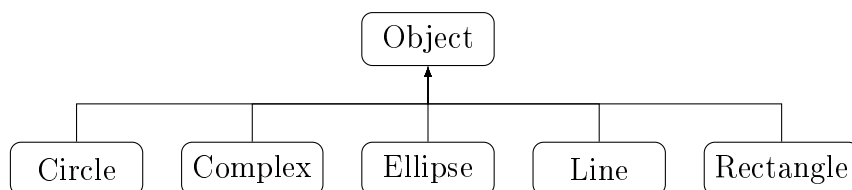
    img.save("test.svg");

    return 0;
}
```

Hierarchia klas

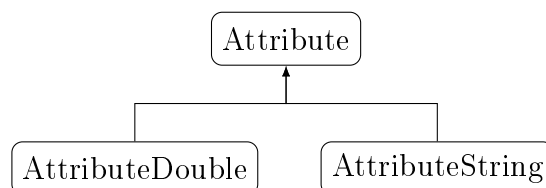
Object

Figury proste i złożone.



Attribute

Cechy figur prostych i złożonych.



Double

Liczba rzeczywista, wymiar, współrzędna, która może być relatywna.

Double

Point

Współrzędne punktu, para (x, y) typu `Double`.

Point