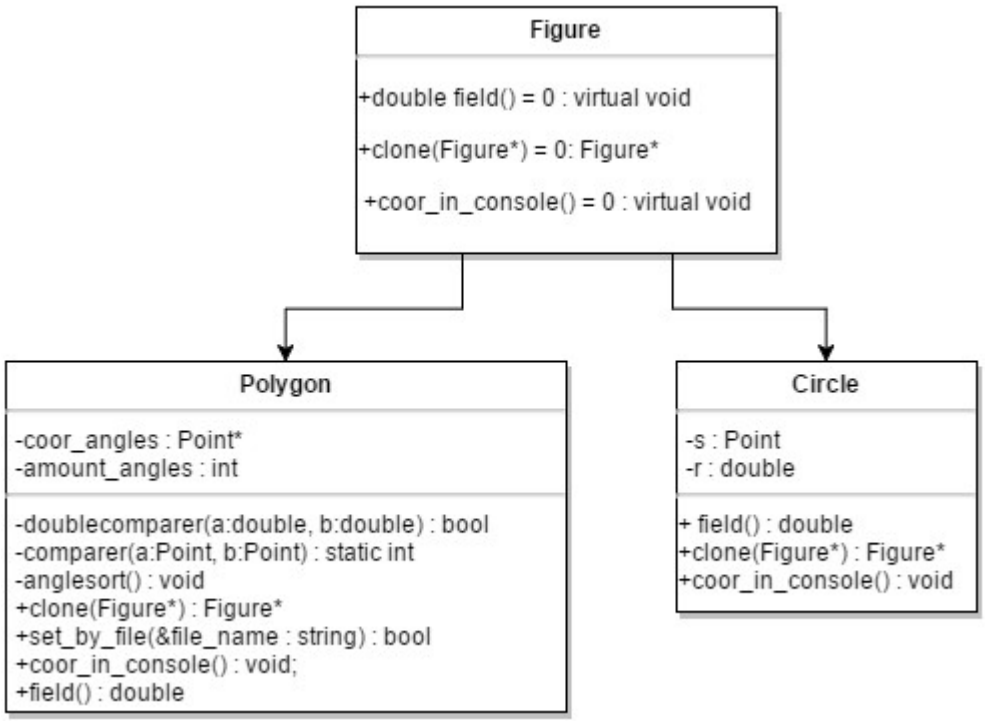
Wstęp

Ze względu na zastosowany autokomentarz nazewniczy, schemat UML poniżej, oraz komentarze opisujące kolejność testów biblioteki w sekcji main, dokumentacja ogranicza się opisu funkcjonalności oraz sposobu przechowywania danych.

Funkcjonalność

Biblioteka operacji geometrycznych, opiera się na dziedziczeniu i polimorfiźmie. Jest w pełni rozszerzalna ze względu na użycie funkcji wirtualnych. Główną klasą abstrakcyjną, która tworzy interfejs, jest klasa Figure.



Jak widać na powyższym diagramie UML, biblioteka umożliwia użytkownikowi na tworzenie instancji dwóch klas – wielokąta oraz koła. Choć liczba klas jest mała, to jednak w pełni wystarczająca aby ukazać zalety polimorfizmu. Widać to na przykładzie klas Plane i PlaneSet, gdzie przechowywane są kolejno figury i płaszczyzny. W ramach podstawowych operacji, zaimplementowana została operacja dodawania na wielokątach (jest to algorytm otoczki wklęsłej), oraz obliczania pola. W kwestii użytkownika pozostaje dopisanie klas dziedziczących np. po Polygon. Miałoby to służyć np. optymalizacji metody obliczania pola.

Sposób przechowywania

Biblioteka umożliwia na przechowywanie figur na płaszczyznach, dzięki klasie Plane oraz na przechowywanie płaszczyzn w zbiorze płaszczyzn, dzięki klasie PlaneSet. Obie klasy działają w oparciu o bibliotekę <vector>. W zależności od preferencji użytkownika, tworzona jest dynamiczna tablica o typie danych dziedziczącym po klasie Figure. Przechowywanie opiera się na kopiowaniu, co dla dużej ilości danych może być obciążające dla pamięci, jednak jest to według autora, metoda lepsza od przechowywania samych wskaźników do figur i płaszczyzn. Umożliwiają to przeciążone operatory przypisania oraz konstruktory kopiujące. Użytkownik biblioteki może dowolnie dodawać i usuwać figury i płaszczyzny ze swoich instancji.