

Zadanie 1.

Zdefiniuj klasę Style o finalnych, publicznych polach klasy String: fillColor, strokeColor oraz Double: strokeWidth. Napisz konstruktor trójargumentowy ustawiający te wartości. Należy dopuścić ustawienie polom wartości null. Napisz publiczną metodę toSvg() zwracającą napis, który można umieścić w opcji style tagu <polygon>.

Zmodyfikuj klasę Polygon dodając do jej konstruktora argument Style i modyfikując jej metodę toSvg(), aby uwzględniała styl. Jeżeli mają wartość null, należy narysować przezroczysty (fillColor) wielokąt, z czarnym obrysem (strokeColor) o grubości jednego piksela (strokeWidth).

Zadanie 2.

Napisz klasę SvgScene posiadającą prywatną tablicę obiektów Polygon. Napisz metodę, która przyjmuje obiekt klasy Polygon oraz dodaje go do tablicy w obiekcie SvgScene. Napisz funkcję saveHtml(String), która utworzy plik HTML w ścieżce danej argumentem i zapisze do niego reprezentacje wszystkich wielokątów znajdujących się na kanwie.

Zadanie 3.

Napisz publiczną, statyczną metodę fabrykującą klasę Polygon o nazwie square. Funkcja powinna przyjąć jako argumenty: obiekt Line, obiekt Style i zwrócić wielokąt będący kwadratem, którego przekątną jest dany odcinek.

Przeciąż metodę klasy Line zwracającą prostokątny odcinek tak, aby przyjmowała jako dodatkowy argument długość zwracanego odcinka.

Zadanie 4.

Utwórz klasę abstrakcyjną Shape, która otrzyma jako pole obiekt klasy Style. Uczyni pole tego obiektu chronionym. Utwórz publiczny konstruktor, który ustawia to pole. Napisz czysto wirtualną metodę toSvg(). Zmodyfikuj klasę Polygon, aby dziedziczyła po klasie Shape.

Zadanie 5.

Napisz klasę Ellipse dziedziczącą po Shape, posiadającą prywatne pola: środek elipsy (Point), długości promieni i styl. W jej implementacji metody toSvg() powinno znaleźć się rysowanie z użyciem tagu <ellipse>.

Zmodyfikuj klasę SvgScene, aby posiadała tablicę obiektów klasy Shape i korzystając z polimorfizmu zapisz w niej obiekty typu Polygon i Ellipse.