

1 Treść projektu

Utwórz klasę `MyPoint` wraz z prywatnymi polami typu `float x` i `float y`. Przygotuj niezbędne konstruktory i metodę `float[] [] getMatrix()` dostarczającą wartości pól w

postaci dwuwymiarowej tablicy o jednej kolumnie i 3 wierszach, jak na przykładzie: $\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

Ponadto klasa ta będzie zawierała metodę `void show()` wypisującą na ekranie nazwę klasy i wartości pól w postaci: `MyPoint x: 3.14 y:6.28`.

Zdefiniuj klasę `AffineTransform`, z statycznymi metodami:

- `float[] [] getTranslateMat(float vx, float vy)` dostarczającą dwuwymiarową tablicę zawierającą $\begin{bmatrix} 1 & 0 & vx \\ 0 & 1 & vy \\ 0 & 0 & 1 \end{bmatrix}$;
- `float[] [] getRotateMat(float phi)` dostarczającą dwuwymiarową tablicę zawierającą $\begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$ Dla wyliczenia tych wartości można użyć metod *Math.cos* i *Math.sin*;
- `float[] [] getScaleMat(float w, float h)` dostarczającą dwuwymiarową tablicę zawierającą $\begin{bmatrix} w & 0 & 0 \\ 0 & h & 0 \\ 0 & 0 & 1 \end{bmatrix}$;
- `MyPoint mul(float[] [], MyPoint)`, której wynikiem będzie dwuwymiarowa tablica o jednej kolumnie której kolejnymi elementami będą wyniki powstałe jako suma mnożeń elementów z wiersza pierwszego argumentu z elementami pierwszej kolumny drugiego argumentu;

UWAGA: operacje w sumowania i mnożenia należy zrealizować z wykorzystaniem pętli, wpisanie ręczne wszystkich możliwości będzie tożsame z **zerem** z projektu.

Następnie utwórz klasy:

- `Triangle`, definiującą:
 - trzy pola klasy `MyPoint` opisujące trzy wierzchołki trójkąta;
 - jeden konstruktor przyjmujący jako argumenty dwie jednowymiarowe tablice opisujące wierzchołki naszego trójkąta i inicjujący pola współrzędnymi;
 - metodę `show` wyświetlającą wszystkie wierzchołki tego trójkąta;
 - metodę `void translate(float vx, float vy)` wykorzystującą metodę `mul` klasy `AffineTransform` aby obliczyć nowe wartości współrzędnych dla każdego z punktów tej klasy;
 - metodę `void rotate(float phi)` wykorzystującą metodę `mul` klasy `AffineTransform` aby obliczyć nowe wartości współrzędnych każdego z punktów tej klasy;

- metodę `void scale(float w, float h)` wykorzystującą metodę `mul` klasy `AffineTransform` aby obliczyć nowe wartości współrzędnych każdego z punktów tej klasy;
- `Circle`, definiującą:
 - domyślny konstruktor tworzący zbiór punktów rozmieszczonych na okręgu w 10 równych odstępach, przyjmij że środek koła jest w punkcie `0,0` a promień jest równy `5`;
 - konstruktor z argumentem `int` opisujący promień okręgu i tworzący zbiór punktów rozmieszczonych na okręgu w 12 równych odstępach, przyjmij że środek koła jest w punkcie `0,0`;
 - metodę `show` wyświetlającą wszystkie wierzchołki tego koła;
 - metodę `void translate(float vx, float vy)` wykorzystującą metodę `mul` klasy `AffineTransform` aby obliczyć nowe wartości współrzędnych dla każdego z punktów tej klasy;
 - metodę `void rotate(float phi)` wykorzystującą metodę `mul` klasy `AffineTransform` aby obliczyć nowe wartości współrzędnych każdego z punktów tej klasy;
 - metodę `void scale(float w, float h)` wykorzystującą metodę `mul` klasy `AffineTransform` aby obliczyć nowe wartości współrzędnych każdego z punktów tej klasy;

Działanie programu wykorzystującego wszystkie zdefiniowane metody przedstaw na przykładzie.

2 Wymagania formalne

Projekt należy oddać do dnia 19.I.2020.

Prace oddawane w ramach projektu powinny być dostarczone w postaci pojedynczego pliku .java o strukturze zgodnej z poniższym schematem:

```
1 public
2     class SXXXXXX_p02 {
3
4         public static void main(String[] args) {
5             // ... tresc programu testowego ...
6         }
7
8         // zestaw metod własnej implementacji studenta
9     }
10
11 class A{
12     // zestaw metod własnej implementacji studenta
13 }
14
15 class B{
16     // zestaw metod własnej implementacji studenta
17 }
```

gdzie jako *XXXXXX* należy podać numer studenta, a nazwy *A* i *B* są przykładowymi nazwami klas.

W projekcie nie można używać żadnych metod i struktur danych, które nie są autorstwa oddającego lub nie zostały wyszczególnione w treści projektu.

Projekt nie może zostać poddany żadnym formom zaciemniania kodu.

Niespełnienie któregośkolwiek z wymogów formalnych opisanych powyżej, skutkować będzie **zerową** liczbą punktów.