

Zadanie jest za 0.4 pkt.

Termin: **3:00, 23.10.2024**

Treść: Zaimplementowane w języku C++ programu, który rysuje figury 2D na płaszczyźnie liczb naturalnych, tj. pozycje i wymiary są typu uint.

Głównym celem jest poprawne zbudowanie **architektury** programu.

W zadaniu można wprowadzić inne, dowolne klasy i metody, należy jednak pamiętać o KISS i SOLID.

Definicje klas i metod są przedstawione w pseudokodzie i należy je zaadaptować odpowiednio do używanego języka (bez zmiany nazw!).

Dodatkowe wymaganie: zadanie powinno być w jednym pliku, wiem, że to nie jest dobra praktyka, ale ułatwia sprawdzanie ;)

Wymagane klasy:

Point:

x: int

y: int

Line: **tylko poziome, pionowe i skośne 45***

set_start(point: Point)

set_stop(point: Point)

is_point() -> bool

Figure: klasa bazowa dla figur

Square: kwadrat, przykład 3x3

```
###
```

```
#  #
```

```
###
```

Rectangle: Prostokąt, chyba oczywiste.

QuarterSquareTriangle: (ćwierć kwadratu: punkty (0,0), (4, 0), (3, 3))

```
#####
```

```
  #  #
```

```
    #
```

HalfSquareTriangle: (pół kwadratu) np.

```
#
```

```
##
```

```
#  #
```

```
####
```

Canvas: Na płótnie można tylko rysować, nie można zmieniać już jego kształtów, ani wymazywać. To co jest narysowane później przesłania poprzednie rysunki. Można rysować poza płótnem, ale wtedy nic się nie zmienia. W naszym przypadku płótno to standardowe wyjście. Płótno ma pozycję (0,0) w lewym górnym rogu, a współrzędne są dodatnie.

Każdorazowa zmiana canvasa powinna wypisywać całe płótno na standardowe wyjście.

Wymagane metody *Figure*:

1. `draw(canvas: Canvas)`: rysuje obiekt na canvasie
2. `set_height(height: uint)`: ustawia wysokość
3. `get_height()` – `> uint`: zwraca wysokość
4. `set_width(width: uint)`: ustawia długość podstawy
5. `get_width()` – `> uint`: zwraca długość podstawy
6. `get_area()` – `> float`: zwraca pole powierzchni obiektu
7. `set_color(c: char)`: ustawia wypisywany znak, np. `#`
8. `get_position()` – `> Point`: `(min(x), min(y))`
9. `move(vector: Line)`: translacja o wektor