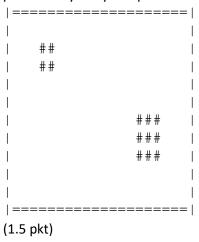
## Zestaw 5

## Zadanie ASCII renderer

Zaprojektuj i napisz program, który będzie tworzył obrazki z formie plików ASCII.

- 1. W pierwszym kroku należy zaimplementować klasę reprezentującą "canvas" na którym będzie powstawał obrazek. Klasa Canvas może być klasą abstrakcyjną, z której będzie dziedziczyła klasa ASCIICanvas. Taka konstrukcja pozwoli na dodanie w przyszłości innych rodzajów canvasów. Canvas powinien mieć możliwość zdefiniowania m.in. wysokości oraz szerokości. W przypadku pliku ASCII to będzie liczba linii i liczba znaków w każdej linii. (0.5 pkt)
- 2. W drugim kroku musisz zdefiniować kształty, które chcesz rysować, np. koło, kwadrat, prostokąt, trójkąt, itp. (minimum 3). Stwórz abstrakcyjną klasę Figure, z której będą dziedziczyły konkretne figury. Możesz także zastosować dziedziczenie wielokrotne, czyli np. z klasy Figure może dziedziczyć klasa Rectangle, a z Rectangle może dziedziczyć klasa Square. Każda figura musi posiadać współrzędne np. środka oraz wymiary pozwalające ją narysować, np. promień w przypadku koła. Każda figura powinna też posiadać funkcję odpowiedzialną za narysowanie jej na canvasie, np. draw(Canvas &c). Poszczególne piksele obrazka powinny być "wypełniane" różnymi symbolami, w zależności od tego, czy należą do obszaru figury czy nie. Możemy np. założyć, że każdy "piksel" obrazka, który należy do figury to #, a każdy piksel, który nie należy do figury to spacja. To jakie symbole będą wykorzystane powinno być ustalane w pliku konfiguracyjnym (patrz kolejny podpunkt) (2 pkt)
- 3. W pliku wykonywalnym zademonstruj rysowanie figur. Niech argumentem funkcji main będzie nazwa pliku konfiguracyjnego. W pliku konfiguracyjnym sprecyzuj następujące rzeczy: wymiar canvasu, nazwę pliku wyjściowego z obrazkiem, symbol oznaczający piksel nienależący do żadnej figury, listę figur do narysowania zawierającą nazwę figury, współrzędne, wymiary, symbol. Plik konfiguracyjny należy sprsować i na jego podstawie stworzyć odpowiednie obiekty i wypełnić obrazek. Jako wynik powinien powstać plik ASCII z naszym obrazkiem. Przykład: Jeśli założyliśmy, że pusty canvas to plik ASCII wypełniony spacjami a figura jest wypełniona # to dla canvasu o wymiarach 10x20 z narysowanymi dwoma kwadratami o boku 2x2 i 3x3 powinniśmy otrzymać plik ASCII o następującej zawartości:



- 4. Projekt umieść w repozytorium github. W ramach rozwiązania tego zestawu, zamiast spakowanych plików źródłowych prześlij link do tego repozytorium. Zadbaj o to, by README.md dołączone do projektu wyjaśniało czego dotyczy projekt, jak go ściągnąć, jak go skompilować i jak go używać. Do stworzenia pliku README.md skorzystaj z języka markdown (<a href="https://www.markdownguide.org/">https://www.markdownguide.org/</a>). Przykładowy plik konfiguracyjny również załącz do repozytorium. (1 pkt)
- 5. \*\*\* podpunkt dodatkowy: Stwórz klasę reprezentującą punktowe źródło światła świecące jednorodnie we wszystkich kierunkach na canvasie (Light). Obiekt Light powinniśmy móc umieścić w dowolnym miejscu na naszym canvasie (możliwość ustawienia współrzędnych). Po umieszczeniu źródła światła na naszym canvasie powinien się pojawić cień rzucany przez figury. Znak symbolizujący cień oraz położenie źródło światła

powinno być określone w pliku konfiguracyjnym. Wracając do powyższego przykładu, po dodaniu źródła światła powinniśmy otrzymać:

	=====
##	*
##	
.	
.#	###
#	###
l#	‡##
· · · · · ·	
· · · · · ·	
=========	=====

(źródło światła oznaczono przez \*, cień oznaczono przez .) (3 pkt)