

## TECHNOLOGIE KOMPONENTOWE GRUPA 2 - 20.11.2018

### 1. Sprawdzenie środowiska pracy:

- a. git --version
- b. dotnet --version
- c. Visual Studio Code

### 2. Fork projektu:

<https://gitlab.com/lukasikpiotr/figures2>

Po ukończeniu zadania proszę o wykonanie Merge Request.

### 3. Stworzyć projekt Classlib w .Net oraz zaimplementuj klasy pozwalające reprezentować zbiór figur geometrycznych występujących w przestrzeni dwuwymiarowej wykorzystując dziedziczenie. (10 pkt)

#### a. Stwórz następujące klasy

- i. Punkt - powinien posiadać dwie wartości rzeczywiste a,b oraz nazwę
- ii. Odcinek - składa się z dwóch punktów oraz nazwy
- iii. Koło - posiada współrzędne środka (jako punkt na płaszczyźnie) oraz długość promienia i nazwę
- iv. Trójkąt - składa się z 3 punktów, posiada nazwę
- v. Czworokąt - składa się z 4 punktów, posiada nazwę
- vi. Wielokąt - posiada listę wierzchołków oraz nazwę

#### b. Każda klasa musi posiadać:

- i. konstruktor parametryczny i domyślny
- ii. metodę pozwalającą na kopiowanie obiektu (implementacja interfejsu ICloneable),
- iii. metodę zwracającą pole figury (jeśli to możliwe),
- iv. metodę zwracającą obwód figury (jeśli to możliwe),
- v. metodę IsValid(), sprawdzającą czy dana figura jest poprawna
  - 1. Jeżeli figura nie jest poprawna (np. punkty leżą na tej samej prostej w przypadku wielokąta) należy wyrzucić własny wyjątek: *InvalidFigure*
- vi. metodę Move(int value), przesuającą figurę o wskazaną wartość
- vii. przeciążoną metodę ToString(), zwracającą nazwę, punkty, obwód i pole,
- viii. metodę ToJson(), zwracającą w.w. dane w postaci JSON,
- ix. przeciążoną metodę Equals() - figury tego samego typu oraz o tych samych właściwościach mają być równe,
- x. przeciążone operatory < i >, w celu porównywania pola powierzchni figur (jeśli to możliwe)

#### c. Dodatkowe operacje:

- i. Klasa punktu powinna posiadać operacje umożliwiające zmianę wartości a i b

- ii. Klasa wielokąta powinna posiadać metodę umożliwiającą dodawanie punktów do listy oraz ich usuwania ze sprawdzeniem czy ta operacja może zostać wykonana
  - 1. Jeśli operacja nie może zostać wykonana, należy wyrzucić stworzony przez Państwa wyjątek: *InvalidDecreaseOperation*
- d. Klasy muszą implementować interfejs:
  - i. IPhoto, zawierający metody:
    - Print() - dowolna implementacja określająca jaka figura jest drukowana,
    - Resize(int procent) - implementacja pozwalająca zmniejszyć figurę o wskazany procent (zmiana położenia punktów, pola powierzchni oraz obwodu)
- 4. Stworzenie projektu z testami (NUnit) sprawdzającymi: (5pkt)
  - a. Stworzenie poprzez konstruktor parametryczny
    - i. Punktu
    - ii. Koła
    - iii. Trójkąta
    - iv. Kwadratu
    - v. Wielokąta
  - b. Kopiowanie wielokąta i sprawdzenie identyczności metodą Equals().
  - c. Przesunięcie obiektu metodą Move() i sprawdzenie nowego położenia.
  - d. Stworzenie błędnego obiektu trójkąta i sprawdzenie metodą IsValid(), a następnie oczekiwanie wyjątku: *InvalidFigure*
  - e. Metody ToString() oraz ToJson().
  - f. Wystąpienie wyjątku *InvalidDecreaseOperation* przy próbie usunięcia zbyt wielu punktów z wielokąta.

5.