

## ZPR PWr – Zintegrowany Program Rozwoju Politechniki Wrocławskiej

PWr

Spotkanie 7

*Aplikacje webowe na platformie .NET*Laboratorium – **Lista 7**

### Wstęp.

Działanie wybranych elementów obiektowych języka C#.

### List zadań

1. Napisać klasę `MixedNumber`, do pamiętania liczby mieszanej, która w czterech właściwościach będzie przechowywała znak (zn), część całkowitą (c), licznik (l) i mianownik (m). Część całkowita (c) i licznik (l) mają być nieujemne, mianownik zawsze dodatni. Dla znaku należy wykorzystać właściwość automatyczną, dla części całkowitej, licznika i mianownika właściwości, które w przypadku podania wartości niepoprawnej wstawi domyślną. Wartość ułamka powinna być **zawsze** w postaci uproszczonej do najmniejszego możliwego mianownika, który zawsze jest większy niż licznik (również po zmianie mianownika lub licznika). Zaproponować minimum 3 konstruktory (część powinna wywoływać inny konstruktor). Przeładować operator '+' i tylko taka operacja matematyczna będzie możliwa. Klasa ma posiadać właściwość do pobierania ułamka jako liczby typu `double`. Zaimplementować własną metodę `ToString()` dla tej klasy. Klasa ma również liczyć ile razy (globalnie) należało uprościć reprezentację obiektów `MixedNumber`, aby zmodyfikować go do postaci zgodnej z założeniem. Ma też, oczywiście, udostępniać możliwość pobrania wartości tego licznika. Napisać dekonstruktor do pobierania wszystkich 4 składowych w jednej instrukcji (zademonstrować działanie).

Przykładowe reprezentacje (zn, c, l, m):

0 -----(1, 0, 0,1)

10/4 ----- (1,2,1,2)

-2/3 -----(-1,0,2,3)

-10/4 -----(-1,2,1,2)

Założenie: używać typ `int` do reprezentacji części całkowitej, licznika i mianownika.Zakładamy, że nie będzie sytuacji, że w wyniku operacji matematycznych wykrócymy poza zakres tego typu. Reprezentacja znaku do własnego uznania (może być np. `bool`). Oznaczenia w czterech pól może być inne niż w treści zadania.

2. Napisać interfejs `IFigure` posiadający metodę:

**void** moveTo(**double** x, **double** y)Oraz interfejs `IHasInterior` z właściwością do ustawienia i pobrania koloru wnętrza. Stworzyć klasy, z których jedne implementują jeden, a drugie obydwa interfejsy. Dla pierwszego interfejsu użyj implementacji niejawnej, dla drugiego implementacji jawnej. Stworzyć tablicę/kolekcję obiektów klasy **object**. Stworzyć

metodę, która otrzyma taką tablicę/kolekcję i dla obiektów implementujących interfejs `IHasInterior` wypisze ich kolor, a dla pozostałych tekst „no color”. Kolor może być typu **string**.

3. Napisać funkcję rozszerzającą klasę **string** o możliwość zamiany ciągu znaków w ten sposób, że pozycje parzyste ciągu będą zamienione na wielkie litery, a nieparzyste na małe. Pozycje nie będące literami mają zostać zamienione na kropkę ‘.’.

Data I: Spotkanie 8 (100 punktów)

Data II: Spotkanie 9 (80 punktów)

Data III: Spotkanie 10 (50 punktów)