

# Lista zadań PSiO 2

Klasy i metody

Termin: 16.11.18

## Cel

- Zapoznanie z klasami, metodami
- Zapoznanie z podstawowymi elementami biblioteki standardowej

## Materiały

- [Java Math](#) - dokumentacja klasy Math zawierających narzędzia wspomagające obliczenia.

## Zadanie

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} \quad \sin(x) = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!} \quad \cos(x) = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k}}{(2k)!}$$

- Utwórz nowy projekt
- W pakiecie domyślnym należy utworzyć klasę `Lab02`, zawierającą jedynie funkcję `main` - punkt startowy projektu.
- W pakiecie `series` należy utworzyć trzy klasy, które reprezentują powyższe rozwinięcia wzorów w szeregi Taylora.
- Każda z klas zawiera przeciążone konstruktory, które umożliwiają podanie:

1. **Argumentu aproksymowanej funkcji (x), Liczby iteracji (n)**
  2. **Argumentu aproksymowanej funkcji (x)**
  3. Żadnych argumentów - Konstruktor domyślny (bezargumentowy), który zawiera wartości domyślne (np.  $x = 2$ ,  $n = 100$ ).
- Każda z klas zawiera dwie metody:
    1. **compute** - wyliczającą wartość danego szeregu dla danych argumentów.
    2. **verbose**, które realizuje to przy pomocy pomocniczych funkcji, zdefiniowanych w osobnej klasie jako metody statyczne, dzięki którym łatwiejsze jest odczytanie pierwotnego wzoru szeregu (np. **power(x, k)**, **factorial(k)**) - możesz w danych metodach statycznych bezpośrednio odwoływać się do biblioteki **Math** w Javie.
  - W metodzie **main** zademonstruj działanie przynajmniej jednego z szeregów (wynik w konsoli).
  - Przygotuj dla prowadzącego wady i zalety obu podejść (**verbose** i **compute**).

Dla chętnych (nieobowiązkowe):

- Utworzyć interfejs dla klas reprezentujących szeregi zawierający deklaracje metod **verbose** i **compute**. Klasy szeregów wtedy powinny implementować ten interfejs.
- \*Czy jesteś w stanie napisać funkcję **sum(k, n, <funkcja obliczająca dany element szeregu>)** (podpowiedź - należy

zastosować dziedziczenie).