

LAB 13 - czw. 14:40

1. Proszę napisać abstrakcyjną klasę *Calka* z metodą inicjalizacyjną określającą granice całkowania, liczbę kroków oraz funkcję podcałkową (proszę skontrolować poprawność przekazanych parametrów) oraz metodą abstrakcyjną obliczającą wartość całki.
Następnie proszę utworzyć klasy dziedziczące po klasie *Calka* z metodami obliczającymi wartość całki odpowiednio metodą trapezów lub Simpsona, w metodzie proszę umieścić komentarz dokumentacyjny. Potrzebne wzory są w pliku: calki.pdf (3.(3)p)
2. Proszę napisać klasę implementującą stos, klasa ma obsługiwać możliwość tworzenia pustego stosu bądź inicjalizacji istniejącym stosem (obiektom klasy), dodawania i usuwania elementu, dodawania elementów innego stosu, zwracania rozmiaru i wypisywania stosu.
Następnie proszę napisać klasę dziedziczącą po klasie stosu i implementującą stos posortowany (rosnąco lub malejąco). W tym przypadku element/elementy innego stosu można do stosu dodać pod warunkiem zachowania porządku sortowania.
Proszę sprawdzić jaki jest średni rozmiar posortowanego stosu, który wypełniamy całkowitymi liczbami losowymi z przedziału [0,100] losując 100 wartości (średnia po 100 powtórzeniach) (3.(3)p)
3. Proszę zaimplementować klasę pozwalającą na zliczanie linii, słów i znaków w pliku (metody inicjalizująca i zliczająca). W klasie proszę także zaimplementować bezparametrową metodę statyczną zwracając komunikat analogiczny do komunikatu zwracanego przez polecenie systemowe linuxa `wc` w przypadku jednoczesnego zliczania dla kilku plików (3.(3)p)

Przykład:

```
$wc AA.py BB.py
```

```
50  91  944 AA.py
```

```
80 117 1281 BB.py
```

```
130 208 2225 razem
```