PWr Spotkanie 01
Algorytmy i struktury danych,
Ćwiczenia – **Lista 01**

Wstęp.

Podczas 45-minutowych zajęć przedstawione zostaną informacje organizacyjne oraz przećwiczone zostanie rozwiązywanie problemów z wykorzystaniem Zoom (z Whiteboard) i Eclipse. W zadaniach są podawana tylko nazwy funkcji. Jakie są ich parametry i zwracane wartości należy wywnioskować z treści zadania.

Lista zadań

1. Zaimplementuj metodę `NextPascalLine()`, która pobiera poprzednią linię w trójkącie Pascala (jako tablicę liczb) i generuje następną linię (jako nową tablicę). Na przykład dla wejściowej tablicy `[1,4,6,4,1]` generuje tablicę `[1,5,10,10,5,1]`. Użyj tej procedury w taki sposób, aby wygenerować cały trójkąt Pascala do danej głębokości wejściowej zaczynając od tablicy [1].
2. Napisz statyczną metodę `getSecondSmallest()`, która otrzymuje tablicę liczb całkowitych i zwraca drugą **najmniejszą wartość** z tej tablicy. Dodatkowym założeniem jest to, że tablica jest traktowana jako strumień danych, więc analizując kolejne wartości w komórkach tablicy, **nie ma możliwości ponownej analizy** tych wartości. Poza prostymi zmiennymi nie wolno tworzyć żadnych dodatkowych struktur (w tym jakiejkolwiek tablicy lub innej kolekcji). Dla strumienia danych (przedstawionego jako tablica) 9, 3, 5, 4, 7, 1, 5, 1, 9 odpowiedzią jest wartość 3, ponieważ tabela ma wartości 1, 3, 4, 5, 7, 9 a wartość 3 jest drugim najmniejszym z nich. Jeśli nie ma poprawnej odpowiedzi - zrzuć wyjątek `NoAnswerException`.
3. Zaimplementuj procedurę `nextPermutation()`, która pobiera wejściową permutację **różnych liczb** (jako tablicę) i generuje następną leksykograficznie permutację (w miejscu, czyli w **tej samej tablicy**). Jeśli taka permutacja nie istnieje, procedura zwraca fałsz, w przeciwnym razie prawda. Spróbuj zaimplementować to efektywnie. Na przykład:
    a. Wszystkie permutacje 3-elementowe w porządku leksykograficznym: 1,2,3 –1,3,2 –2,1,3 –2,3,1 –3,1,2 –3,2,1
    b. Kilka kolejnych permutacji po 3,4,2,5,1 to 3,4,5,1,2 –3,4,5,2,1 -3,5,1,2,4