Praca domowa nr 2
(Odwrotna Notacja Polska)

Należy napisać pełny projekt w Javie, który będzie zawierał program umożliwiający obliczanie wyrażeń nawiasowych na liczbach całkowitych z użyciem podstawowych działań arytmetycznych: +, **~(minus)**, *, /, % (modulo), ^ (potęgowanie) oraz par nawiasów: (,).

Dodatkowo, w celu wykorzystania w algorytmach związanych z ONP, definiujemy priorytety operatorów:

Priorytet	Operatory
0	(
1	+ ~)
2	* / %
3	^

W szczególności program powinien zawierać własne implementacje klas:

- **kolejka FIFO** (elementów typu podstawowego String lub Object w zależności od uznania twórcy programu)
- **stos LIFO** (elementów jak w kolejce)

oraz definiować następujące operacje (zamknięte w odpowiednio zadeklarowanych funkcjach):

Konwersja wyrażenia nawiasowego na ONP

(wejściem do funkcji jest kolejka FIFO typu String lub Object, wyjście tego samego typu)
Opis Algorytmu w krokach:

- utwórz roboczy Stos (początkowo pusty)
- Dopóki w kolejce wejściowej są jeszcze symbole (operatory lub liczby), powtarzaj:
  - pobierz jeden element z kolejki wejściowej i w zależności od jego rodzaju wykonaj
    - jeśli symbol jest liczbą dodaj go do kolejki wyjściowej (ONP)
    - jeśli symbol jest nawiasem otwierającym „(” , to połóż go na stos
    - jeśli symbol jest operatorem
      1. połóż go na stos o ile na szczycie stosu nie znajduje się operator o priorytecie większym lub równym (op. wolno położyć na stos pusty lub na operator o niższym priorytecie)
      2. w przeciwnym wypadku zdejmuj ze stosu operatory i odkładaj do kolejki wyjściowej ONP, aż będzie możliwe 1.
    - jeśli symbol jest nawiasem zamykającym „)”, to zdejmuj operatory ze stosu i dokładaj do kolejki wyjściowej ONP, aż zdejmiesz nawias otwierający (nawiasu nigdzie nie odkładaj, tylko skończ operację, ponieważ konwertujemy do wyrażenia beznawiasowego)
- **Gdy kolejka wejściowa będzie pusta - przepisz wszystkie operatory ze stosu do kolejki wyjściowej (nie powinno tam być żadnych nawiasów)**

Obliczenie wartości wyrażenia zapisanego w formacie ONP

(wejściem do funkcji jest kolejka FIFO typu String lub Object, wyjściem pojedyncza liczba całkowita)

Opis Algorytmu w krokach:

- utwórz roboczy Stos (początkowo pusty)
- Dopóki w kolejce wejściowej są jeszcze symbole ONP (operatory lub liczby), powtarzaj:
  - pobierz jeden element z kolejki wejściowej i w zależności od jego rodzaju wykonaj
    - jeśli symbol jest liczbą, odłóż go na stos
    - jeśli symbol jest operatorem (dwuargumentowym, bo innych nie rozpatrujemy)
      - zdejmij ze stosu jeden element (ozn. ***a***)
      - zdejmij ze stosu drugi element (ozn. ***b***)
      - wykonaj działanie ***b operator a*** i wynik odłóż na stos
- Zdejmij ze stosu wynik wyrażenia (jeśli wyrażenie było poprawne na stosie powinien się znajdować pojedynczy element będący liczbą)

Konwersja wyrażenia napisowego na kolejkę

(wejściem do funkcji jest napis typu String, wyjściem kolejka elementów odpowiedniego typu)

Wskazówka: napis najprościej „pociąć” na znakach białych wykorzystując klasę StringTokenizer.

Program powinien działać następująco:

1. Poprosić użytkownika o wyrażenie w postaci nawiasowej, gdzie (dla uproszczenia) każdy element wyrażenia jest rozdzielony znakiem białym (spacją) i pobrać to wyrażenie
2. Przekonwertować napis zawierający wyrażenie nawiasowe do wyrażenia nawiasowego w postaci kolejki FIFO odpowiedniego typu (użyć wcześniej napisanej funkcji)
   - można tę kolejkę testowo wyświetlić na ekran
3. Zamienić kolejkę z wyrażeniem nawiasowym na kolejkę w formacie ONP (użyć wcześniej napisanej funkcji)
   - po tej konwersji **należy** wyświetlić kolejkę na ekran
4. Obliczyć wartość wyrażenia ONP zapisanego wcześniej jako kolejka (użyć wcześniej napisanej funkcji) i wyświetlić tę wartość na ekranie

Program **powinien** wychwytywać jak najwięcej błędów w wyrażeniach i w przypadku błędu wyświetlać odpowiedni komunikat (można przerwać standardowe działanie wyjątkiem)

Przykłady działania poszczególnych algorytmów związanych z wyrażeniami ONP można znaleźć m.i. na wikipedii: http://pl.wikipedia.org/wiki/Odwrotna_notacja_polska