

## Zajęcia C-3: “Stos, Odwrotna Notacja Polska”

### Cel zajęć i efekty uczenia

Główne cele zajęć / materiał do opanowania:

- Struktura stosu, szablon *stack* w STLu
- Odwrotna Notacja Polska, algorytm obliczania wyrażenia zadanego w ONP (maszyna stosowa)
- [opcjonalnie] Algorytm zamiany na ONP (“algorytm nastawni kolejowej”)

Dodatkowe cele:

- [opcjonalnie] Definicja nawiasowania przez reguły przepisywania (gramatyka formalna)

### Zadania do rozwiązania na sprawdzarce

#### **Nawiasy**

*Dany jest napis złożony ze znaków nawiasów (,),[,],{,}. Rozstrzygnąć, czy jest poprawnym nawiasowaniem.*

#### **Odwrotna Notacja Polska**

*Dane jest wyrażenie w Odwrotnej Notacji Polskiej, złożone z liczb oraz znaków + i \*. Obliczyć jego wartość.*

#### **[opcjonalnie] Odwrotna Notacja Polska kontratakuje**

*Dane jest wyrażenie zapisane w standardowej notacji, z liczbami i znakami +, \*. Zamienić je na Odwrotną Notację Polską.*

Szacunkowy czas trwania: 2-4 godziny lekcyjne.

### Plan zajęć

Szacunkowy czas trwania: 2 godziny lekcyjne (sama kolejka), lub 4 godziny (z techniką obgryzania).

1. Struktura stosu, stos w C++/STL i własna implementacja
  - *Większość tłumaczenia odnośnie struktur danych i szablonów, została już wykonana w przypadku kolejki, stos jest nawet prostszy w implementacji.*
  - *Przy stosie nawet częściej o błąd “wyciągania z pustej struktury”, warto zwrócić na to szczególną uwagę.*
2. Zadanie “Nawiasy”
  - *Zaczynamy od jednego rodzaju nawiasu: co to znaczy, że nawiasowanie jest poprawne?*

- *Możliwa definicja: “wtedy, kiedy nawiasy można dobrać w pary otwierający-zamykający tak, że żadne dwie pary się nie krzyżują”.*
  - *Dla jednego rodzaju nawiasów działa następujący algorytm: idziemy od lewej do prawej, za każdy otwierający nawias dodajemy 1, za zamykający odejmujemy 1. Wyrażenie jest poprawne, jeśli nigdy nie zejdziemy poniżej 0, a na końcu jest dokładnie 0.*
  - *Dla dwóch i więcej rodzajów nawiasów powyższy algorytm nie działa. Warto dać uczniom okazję do poprawiania go samodzielnie - to bardzo dobry moment, żeby wspólnie z nimi wymyślać nowe rozwiązania i wspólnie szukać kontrprzykładów na nie działające.*
  - *Algorytm ogólny: nawiasy otwierające dorzucamy do stosu. Nawias zamykający musi zawsze pasować do ostatniego nawiasu otwierającego na stosie - jeśli nie, wyrażenie jest niepoprawne, jeśli tak, ściągamy ze stosu ten ostatni otwierający.*
  - *Jeśli na końcu stos jest niepusty, wyrażenie da się uzupełnić do poprawnego, kolejno dopisując zamykające nawiasy pasujące do ostatniego otwartego.*
3. *Odwrotna Notacja Polska, algorytm obliczania wyrażenia*
- *Kwestia techniczna: trzeba czytać z wejścia dane, które raz są liczbą, a raz operatorem działania, i rozpoznawać po pierwszym znaku. Liczby następnie zamieniamy na typ liczbowy funkcją `atoi()` lub `stoi()`.*
4. *[opcjonalnie] Odwrotna Notacja Polska, algorytm zamiany na ONP*
- *Warto zacząć od sytuacji, kiedy jest tylko jeden operator (+) i nawiasy. Kiedy już uczniowie rozumieją, jak obsługuje się nawiasy, wprowadza się kwestię priorytetu operatorów.*
  - *Program wymaga dużej ilości instrukcji `if/switch`, a dodatkowo błędy bywają trudne do wykrycia. Może wymagać dużo pracy od uczniów i poprawiającego ich programy nauczyciela.*

