

Laboratorium 1 :

Powtórzenie podstaw programowania w języku Java. Stosy. Proste algorytmy wykorzystujące stosy.

1. Koncepcja stosu.

Stos jest jednym z rodzajów struktur danych. Kolejne elementy (dane) są układane na wierzchołku stosu. Usunięty może być tylko element z wierzchołka stosu. Dobrym przykładem obrazującym działanie stosu jest stos książek na biurku. Możemy dołożyć kolejną książkę kładąc ją na wierzchołku, po czym to właśnie nowo dołożona książka staje się wierzchołkiem stosu. Jeśli chcemy z takiego stosu wziąć książkę, to możemy wziąć tylko książkę, która leży na wierzchu. Jeśli chcemy sięgnąć po inną książkę, musimy kolejno zdejmować książki z wierzchołka.

2. Operacje na stosie.

W zależności od implementacji wymienione operacje mogą różnić się nieco nazwami i funkcjonalnością, ale podstawowe działania na stosie obejmują:

push – dołożenie elementu na stos

pop – ściągnięcie elementu ze stosu

top – zwraca wartość elementu znajdującego się na szczycie stosu

isEmpty – sprawdzenie, czy stos jest pusty

3. Zadania powtórzeniowe. Rozwiąż poniższe zadania dwukrotnie: raz używając pętli, potem z wykorzystaniem rekurencji.

a. Napisz program obliczający sumę cyfr podanej liczby. [2p]

b. Napisz program sprawdzający czy wczytana liczba naturalna jest liczbą pierwszą. [2p]

4. Zadania - stosy.

1. Utworzyć nowy projekt Java w środowisku NetBeans lub Eclipse. Utworzyć pakiet **stos**. Ze strony WWW pobrać interfejs **IStack** i umieścić go w pakiecie **stos**. W pakiecie **stos** utworzyć klasy **StackT** oraz **StackL** implementujące interfejs **IStack**. Wygenerować szkielety metod deklarowanych przez **IStack**. [1p]

2. Zaimplementować klasę **StackT** w oparciu o tablicę. Zaimplementować konstruktor tworzący stos o zadanym maksymalnym rozmiarze i metodę wypisującą zawartość stosu. [2p]

3. Zaimplementować klasę **StackL** w oparciu o strukturę wiązaną. Należy wykorzystać klasę **SElem** (do pobrania ze strony WWW). Zaimplementować metodę wypisującą zawartość stosu. [3p]

4. Napisać program wykorzystujący stos, który sprawdza czy w podanym wyrażeniu nawiasy są prawidłowo zagnieżdżone. Np. wyrażenie "(() (())" jest prawidłowe, a ")(" oraz "((" są nieprawidłowe. [2p]

5. Napisać program wykorzystujący stos, który sprawdza czy podane słowo jest palindromem. [2p]

6. Napisać program czytający wyrażenia arytmetyczne w notacji postfiksowej (Odwrotna Notacja Polska) i obliczający ich wartość z wykorzystaniem stosu. [2p]

Praca domowa:

1. Napisać program czytający z pliku wyrażenia arytmetyczne zbudowane z wykorzystaniem operatorów + - * / () i zamieniający je na wyrażenia w notacji postfiksowej z wykorzystaniem stosu.
2. Zaimplementować stos jako klasę generyczną.
3. Rozwiązać zadania 2-4 wykorzystując klasę `java.util.Stack` lub implementację interfejsu `java.util.Deque`.

Na następnych zajęciach:

Listy. Kolejki. Operacje na listach i kolejkach.