Projektowanie algorytmów i metody sztucznej inteligencji

Przemysław Erbert 258964

wt 15:15

1 Link do repozytorium

https://github.com/Przemoerb/Pamsi1

2 Opis poszczególnych klas

W programie zdefiniowane sa dwie klasy: Element i Stack.

Klasa Element reprezentuje pojedynczy element stosu jako dynamiczna strukture danych, która zawiera wartość typu int oraz wskaźnik na nastepny element (typu Element*). Struktura ta jest używana tylko przez klase Stack.

Klasa Stack reprezentuje abstrakcyjny stos, który może być zaimplementowany jako stos na liście lub stos na tablicy. Klasa ta posiada dwie prywatne zmienne: int* arrayStack oraz Element* listStack. Pierwsza z nich reprezentuje tablice elementów stosu, druga reprezentuje liste elementów stosu. Zmienna arraySize przechowuje aktualna liczbe elementów na stosie.

3 Opis poszczególnych funkcji

Klasa Stack posiada publiczne funkcje, które umożliwiaja operacje na stosie. Funkcje pushElementToList, popElementFromList, clearList i printList umożliwiaja operacje na stosie jako liście. Natomiast funkcje pushElementToArray, popElementFromArray, clearArray i printArray umożliwiaja operacje na stosie jako tablicy. Funkcje pushElementToList i pushElementToArray dodaja element na stos, popElementFromList i popElementFromArray usuwaja ostatnio dodany element ze stosu, clearList i clearArray usuwaja wszystkie elementy ze stosu, natomiast printList i printArray wyświetlaja zawartość stosu na standardowym wyjściu.

- Stack::Stack() Konstruktor klasy Stack inicjuje wskaźniki listStack i arrayStack na wartość nullptr, a arraySize na 0.
- Stack:: Stack() Destruktor klasy Stack zwalnia pamieć zaalokowana na tablice arrayStack oraz elementy listy listStack.

- void Stack::pushElementToList(int value) Metoda pushElementToList dodaje nowy element do listy listStack. Tworzy nowy element, inicjuje jego wartość przekazanym argumentem value, a nastepnie przypisuje go jako nowy szczyt listy.
- int Stack::popElementFromList() Metoda popElementFromList usuwa pierwszy element z listy listStack. Pobiera wskaźnik na szczyt listy, przypisuje jej nastepnik jako nowy szczyt, usuwa pierwotny szczyt go zwraca.
- void Stack::clearList() Metoda clearList usuwa wszystkie elementy z listy listStack. Wywołuje metode popElementFromList aż do momentu, gdy lista bedzie pusta.
- void Stack::printList() Metoda printList wypisuje wszystkie elementy listy listStack na standardowe wyjście.
- void Stack::pushElementToArray(int value) Dodaje element o wartości value na wierzchołek stosu tablicowego. Jeśli tablica jest już pełna, to alokuje nowa tablice o dwa razy wiekszej wielkości, kopiujac elementy ze starej tablicy. Zwieksza wartość top o 1 i dodaje nowy element na wierzchołek tablicy.
- int Stack::popElementFromArray() Usuwa element ze szczytu stosu tablicowego i zwraca jego wartość. Sprawdza czy stos jest pusty, a nastepnie zdejmuje wierzchołek stosu tablicowego i zmniejsza wartość top o 1. Jeśli po usunieciu elementu rozmiar stosu jest mniejszy niż połowa wielkości tablicy, to alokuje nowa tablice o połowe mniejsza, kopiujac elementy ze starej tablicy.
- void Stack::clearArray() Metoda clearArray usuwa wszystkie elementy z tablicy arrayStack. Ustawia rozmiar tablicy na 0.
- void Stack::printArray() Metoda printArray wypisuje wszystkie elementy tablicy arrayStack na standardowe wyjście. Przechodzi przez tablice od ostatniego elementu do pierwszego i wyświetla wartość każdego elementu.

4 Wynik działania programu

Czas pushowania dla 1000000 danych to (tablica): 18 milisekund
Czas pushowania dla 1000000 danych to (lista): 67 milisekund
Czas popowania dla 1000000 danych to (tablica): 7 milisekund
Czas popowania dla 1000000 danych to (lista): 36 milisekund