

ЛАБОРАТОРНА РОБОТА 1.1

ДОСЛІДЖЕННЯ ЛІНІЙНИХ СТРУКТУР ДАНИХ

Мета – дослідження лінійних структур даних залежно від способу розміщення їх елементів у пам'яті та набуття практичних навичок зі створення й обробки лінійних структур даних.

Завдання

Виконати такі дії:

- описати структуру даних (табл. 1.1, кол. 2, 5) з векторним способом розміщення елементів у пам'яті. Тип елементів структури даних обирати згідно з варіантом завдання (табл. 1.1, кол. 3, 6);
- описати структуру даних (табл. 1.2, кол. 2) зі зв'язаним способом розміщення елементів у пам'яті. Тип елементів структури даних обирати згідно з варіантом завдання (табл. 1.2, кол. 3);
- створити екземпляри описаних структур даних;
- вставити елементи до першої структури даних (табл. 1.3, кол. 2) і вивести її вміст;
- сформувати другу структуру даних (табл. 1.3, кол. 4) згідно із завданням (табл. 1.3, кол. 3);
- вивести вміст обох структур даних.

Методичні рекомендації

Структура даних з векторним способом розміщення елементів у пам'яті описується як клас, який містить одновимірний масив і кількість елементів у структурі даних. Крім того, у цей клас необхідно помістити додаткову змінну, таку як вершина стеку, хвіст черги або списку, яка дозволяє виконувати обробку структури даних. Порядок додавання та видалення елементів з масиву має відповідати правилам роботи із заданою у варіанті структурою даних (табл. 1.1).

У класі обов'язково повинні міститися два методи, які виконують спеціальні перевірки. Перший метод (*bool isFull()*) перевіряє наповненість структури даних (чи дорівнює довжина масиву кількості елементів у структурі даних) та викликається в операції вставки для запобігання додавання елемента в цілком заповнену структуру даних. Другий метод (*bool isEmpty()*) перевіряє порожність

структури даних (чи дорівнює нулю кількість елементів у структурі даних) та викликається в операції видалення для запобігання отриманню елемента з порожньої структури даних.

Метод, що реалізує операцію вставки у структуру даних, повинен повертати логічне значення, яке показує успішність цієї операції. Метод, що реалізує операцію видалення зі структури даних, повинен повертати елемент, який видаляється, або генерувати виняткову ситуацію, якщо такий елемент відсутній. Якщо видаляється елемент не з хвоста черги або списку, необхідно передбачити зсув ліворуч на одну позицію всіх елементів, розміщених праворуч від елемента, що видаляється.

Структура даних зі зв'язаним способом розміщення елементів у пам'яті описується як клас, який містить посилання на елемент або елементи, за допомогою яких виконується обробка структури даних. Такими елементами є вершина стека, голова та хвіст черги або перший елемент списку. Не слід у структурі даних зберігати надлишкові дані, такі як кількість елементів структури даних. Тип елементів структури даних описується як клас, який містить дані та посилання на сусідній елемент або елементи залежно від типу структури даних у варіанті завдання (табл. 1.2).

У класі обов'язково повинен міститися спеціальний метод (*bool isEmpty()*), який перевіряє порожність структури даних (чи дорівнює значенню *null* посилання на елемент, за допомогою якого виконується обробка структури даних) та викликається в операції видалення для запобігання отриманню елемента з порожньої структури даних.

Контрольні запитання

1. Що таке структура даних? За якими критеріями можна класифікувати структури даних?
2. Якими способами розміщуються елементи структур даних у пам'яті?
3. Як реалізована структура даних «стек» у разі векторного способу розміщення елементів у пам'яті?
4. Як реалізована структура даних «черга» у разі векторного способу розміщення елементів у пам'яті?
5. Як реалізована структура даних «стек» у разі зв'язаного способу розміщення елементів у пам'яті?

6. Як реалізована структура даних «черга» у разі зв'язаного способу розміщення елементів у пам'яті?
7. Що таке рекурсивна функція? Які особливості її реалізації?
8. У чому полягає особливості рекурсивної реалізації операцій структури даних?

Учебные материалы ИПО НАУ специальности ПОС

Таблиця 1.1 – Опис першої структури (за векторним представленням)

Варіант	Тип структури даних	Тип елементів	Варіант	Тип структури даних	Тип елементів
1	2	3	4	5	6
1	Стек	Строковий (представляє цілі числа)	11	Стек	Цілий
2	Черга	Цілий	12	Черга	Цілий
3	Список	Строковий (представляє цілі додатні числа)	13	Черга	Дійсний
4	Стек	Дійсний	14	Черга	Дійсний
5	Список	Цілий	15	Список	Цілий
6	Черга	Цілий	16	Стек	Цілий
7	Список	Цілий	17	Список	Строковий (представляє цілі додатні числа у шістнадцятковій системі числення)
8	Черга	Строковий	18	Черга	Цілий беззнаковий
9	Список	Цілий	19	Стек	Дійсний
10	Стек	Строковий (представляє цілі додатні числа у вісімковій системі числення)	20	Черга	Цілий

Таблиця 1.2 – опис другої структури (за зв'язаним представленням)

Варіант	Тип структури даних	Тип елементів
1	2	3
1.	Односпрямований список	Цілий
2.	Двоспрямований список	Цілий
3.	Стек	Цілий
4.	Черга	Строковий (представляє цілі додатні числа у вісімковій системі числення)
5.	Двоспрямований список	Строковий (представляє цілі числа)
6.	Двоспрямований список	Строковий (представляє цілі числа)
7.	Стек	Строковий (представляє цілі додатні числа у вісімковій системі числення)
8.	Стек	Перелічувальний (представляє жіночі імена на парних позиціях, чоловічі – на непарних)
9.	Двоспрямований список	Строковий (представляє цілі додатні числа)
10.	Односпрямований список	Цілий
11.	Односпрямований список	Строковий (представляє цілі додатні числа у шістнадцятковій системі числення)
12.	Односпрямований список	Строковий (представляє цілі додатні числа у двійковій системі числення)
13.	Стек	Строковий (представляє цілі додатні числа у двійковій системі числення)
14.	Двоспрямований список	Строковий (представляє цифрові символи)
15.	Черга	Строковий (представляє цілі числа у шістнадцятковій системі числення)
16.	Односпрямований список	Строковий (представляє цілі додатні числа)
17.	Стек	Цілий
18.	Двоспрямований список	Цілий
19.	Односпрямований список	Строковий (представляє цілі числа у діапазоні від -10 до 10)
20.	Односпрямований список	Цілий

Таблиця 1.3 – Завдання

Варіант	Перша структура даних	Завдання	Друга структура даних
1	2	3	4
1.	Стек (див. табл.1.1)	Перемістити елементи стеку до списку у такий спосіб: парні елементи додаються до початку списку, а непарні – у кінець.	Список (табл.1.2)
2.	Список (табл.1.2)	Видалити зі списку додатні елементи та перемістити їх у чергу так: кожен елемент черги обчислюється як сума попереднього елемента черги та щойно видаленого елемента списку.	Черга (табл.1.1)
3.	Список (табл.1.1)	Видалити зі списку елементи, які представляють непарні цілі числа більше за 50, та перемістити їх у стек.	Стек (табл.1.2)
4.	Черга (табл.1.2)	Перемістити всі елементи черги у стек так, щоб елементи черги, перетворені в десяткову систему числення і більші за 200, – зменшувалися вдвічі, а менші – подвоювалися.	Стек (табл.1.1)
5.	Список (табл.1.1)	Видалити зі списку парні числа та вставити їх у другий список у відсортованому за зростанням порядку.	Список (табл.1.2)
6.	Список (табл.1.2)	Видалити зі списку парні числа та перемістити їх у чергу.	Черга (табл.1.1)
7.	Список (табл.1.1)	Видалити зі списку від'ємні елементи, а решту перетворити у вісімкову систему та перемістити до стеку.	Стек (табл.1.2)
8.	Стек (табл.1.2)	Перемістити елементи стеку, які представляють жіночі імена до першої черги, а чоловічі імена – до другої черги.	Черга (табл.1.1)
9.	Список (табл.1.1)	Видалити зі списку парні додатні числа та вставити їх у список у відсортованому за спаданням порядку.	Список (табл.1.2)
10.	Список (табл.1.2)	Видалити зі списку від'ємні елементи, інвертувати їх, перетворити у вісімкову систему числення та перемістити у стек.	Стек (табл.1.1)

Закінчення табл. 1.3

1	2	3	4
11.	Стек (табл.1.1)	Перемістити додатні елементи стеку, перетворені у шістнадцяткову систему числення, до початку списку, а від'ємні елементи стеку інвертувати, перетворити в шістнадцяткову систему числення і перемістити в кінець списку.	Список (табл.1.2)
12.	Список (табл.1.2)	Видалити зі списку парні елементи, а решту елементів скопіювати до черги в десятковій системі числення.	Черга (табл.1.1)
13.	Черга (табл.1.1)	Перемісти до стеку додатні заокруглені елементи черги, перетворені у двійкову систему числення, а від'ємні елементи - видалити.	Стек (табл.1.2)
14.	Список (табл.1.2)	Видалити зі списку елементи, які містять непарну кількість символів, а решту елементів використати для формування черги так, щоб елемент черги обчислювався як середнє арифметичне цифр елемента списку.	Черга (табл.1.1)
15.	Список (табл.1.1)	Видалити зі списку парні елементи, а решту скопіювати до черги у шістнадцятковій системі числення.	Черга (табл.1.2)
16.	Список (табл.1.2)	Перемістити зі списку в стек елементи, які представляють прості числа.	Стек (табл.1.1)
17.	Список (табл.1.1)	Перетворити елементи списку в десяткову систему числення та обчислити кожен елемент стеку як суму поточного, попереднього та наступного елементів списку.	Стек (табл.1.2)
18.	Список (табл.1.2)	Видалити зі списку від'ємні елементи, а решту елементів використати для обчислення елементів черги, які є сумою цифр елемента списку.	Черга (табл.1.1)
19.	Список (табл.1.2)	Перетворити в цілий тип елементи списку і використати їх як степінь, у яку необхідно піднести число 10, щоб отримати значення елементів стеку.	Стек (табл.1.1)
20.	Черга (табл.1.1)	Перемістити елементи черги до списку, вставляючи елементи так, щоб однакові числа розташовувалися в списку один за одним, а перший елемент такої послідовності містив їх кількість.	Список (табл.1.2)

СПИСОК ЛИТЕРАТУРЫ

1. Алгоритмы: построение и анализ, 2-е изд. / Т. Кормен, Ч. Лейзерон, Р. Ривест, К. Штайн. – М. : Издательский дом «Вильямс», 2005. – 1296 с.
2. Гудрич М. Т. Структуры данных и алгоритмы в Java / М. Т. Гудрич, Р. Тамассия; Пер. с англ. А. М. Чернухо. – Мн. : Новое знание, 2003. – 671 с.
3. Кнут Д. Искусство программирования. – Т.1. Основные алгоритмы, 3-е изд. / Д. Кнут. – М. : Издательский дом «Вильямс», 2010. – 720 с.
4. Роберт Седжвик. Фундаментальные алгоритмы на Java. Анализ/Структуры данных/Сортировка/Поиск. / Р. Седжвик. – К. : ООО «ТИД ДС», 2003. – 688 с.

Учебные материалы ИПО НАУ специальности ИРОС