**Дополнительно задание на тему «Простейшие структуры данных»**

Ответьте *РАЗВЕРНУТО* на следующие вопросы. Документ с ответами загрузите в свой репозиторий, после чего создайте коммит с названием «C1 additional complete». Напишите мне в дискорде о выполнении, чтобы получить баллы.

1. Как устроена структура данных «Динамический массив». какой интерфейс она предоставляет. В чем ее преимущества и недостатки.

Динамический массив отличается от статического способностью изменить свой размер во время работы.

Интерфейс включает в себя:

1. get - получение значения элемента по индексу

2. set - установка значения по индексу

3. delete - удаление элемента по индексу

4. insert - добавление элемента в указанную позицию

5. find - поиск элемента по значению

6. resize - изменение размера массива

Преимуществами динамического массива является возможность изменения размера в зависимости от необходимого количества данных, быстрый доступ к элементам.

Недостаток - большое количество времени, уходящее на изменение размера массива.

2. Как устроена структура данных «Двусвязный список». Какой интерфейс она предоставляет. В чем ее преимущества и недостатки.

Двусвязный список состоит из элементов, каждый из которых содержит в себе значение и ссылки на последующий и предыдущий элемент.

Интерфейс:

1. get - получение значения по индексу

2. set - установка значения по индексу

3. getSize - получение размера

4. find - поиск элемента по значению

5. insertBefore - добавление элемента после данного

6. insertAfter - добавление элемента после данного

7. addFirst/addLast - добавление элемента в начало/конец

8. delete - удаление данного элемента

Преимущества по сравнению с односвязным списком - возможность перемещаться в списке в обе стороны, проще производить удаление и перестановку элементов, быстрая операция вставки в сравнении с массивом.

Недостатки - медленный доступ к элементу в сравнении с массивом, большой расход памяти на хранение указателей.

3. Что такое логическая структура данных? Опишите известные вам логические структуры данных. Какой интерфейс они предоставляют, как реализуются?

Логическая структура данных - абстрактное представление структуры данных, созданное программой.

Структуры: Stack - список элементов, организованный по принципу LIFO (last in - first out)

Интерфейс:

1. size, isEmpty

2. push

3. pop

4. peek

Реализация (на списке)

data: список (двусвязный)

top: фиктивный элемент

size, isEmpty делегируется List

push - вызов добавления в конец, обновление top

peek - возвращение данных из top

pop - возвращение данных из top, удаление top из data, обновление top

Queue - список элементов, организованный по принципу FIFO (first in - first out)

Интерфейс:

1. enqueue

2. dequeue

3. peek

4. size

Реализация на списке:

создание пустого списка

enque - добавление элемента в конец списка

deque - удаление элемента из начала списка

peek - возвращает значение head

size - делегируется list

Dequeue (Double ended queue) - очередь, элементы в которую можно добавлять как в начало, так и в конец

Интерфейс:

1.push\_front

2.peek\_front

3.pop\_front

4.push\_back

5.peek\_back

6.pop\_back

Реализация на списке:

push\_front - добавление элемента в начало списка

pop\_front - удаление head, возвращение его значения

peek\_front -возвращение значения head

push\_back - добавление элемента в конец списка

pop\_back - удаление tail, возвращение его значения

peek\_back - возвращение значения tail