

1. В чем концептуальная разница между LinkedList и ArrayList?

LinkedList и ArrayList - это две разные реализации списка в Java, и у них есть несколько концептуальных различий:

Структура данных:

ArrayList: ArrayList представляет собой динамический массив, где элементы хранятся в непрерывной области памяти. Он предоставляет быстрый доступ к элементам по индексу.

LinkedList: LinkedList - это двусвязанный список, где каждый элемент (узел) содержит ссылку на предыдущий и следующий элемент.

Сложность доступа к элементам:

ArrayList: ArrayList обеспечивает быстрый доступ к элементам по индексу ($O(1)$), но медленный доступ по сдвигу ($O(n)$), когда вы вставляете или удаляете элементы в середине списка.

LinkedList: LinkedList обеспечивает быстрый доступ к элементам по сдвигу ($O(1)$), но медленный доступ по индексу ($O(n)$), так как вам нужно пройти через элементы, начиная с начала списка или с конца.

Вставка и удаление:

ArrayList: Вставка и удаление элементов в середине списка (по индексу) требуют сдвига всех последующих элементов, что делает операции вставки и удаления медленными ($O(n)$).

LinkedList: Вставка и удаление элементов в середине списка (по сдвигу) происходит быстро ($O(1)$), так как нужно только обновить ссылки на соседние узлы.

Память:

ArrayList: ArrayList в целом требует меньше памяти, чем LinkedList, так как он хранит только массив элементов и может быть более компактным.

LinkedList: LinkedList требует дополнительной памяти для хранения ссылок на узлы, что может занять больше места.

2. Для каких задач лучше выбрать LinkedList, а когда ArrayList?

Когда выбрать LinkedList:

Если вам часто нужно вставлять или удалять элементы в середине списка.

Если вам необходима быстрая вставка/удаление элементов, но вы готовы пожертвовать быстрым доступом по индексу.

Если ваш список будет относительно небольшим, и вы не беспокоитесь о дополнительной памяти для хранения ссылок.

Когда выбрать ArrayList:

Если вам нужен быстрый доступ к элементам по индексу.

Если вы не часто вставляете или удаляете элементы в середине списка.

Если ваш список будет большим, и вы хотите оптимизировать использование памяти.

3. Какую информацию хранит каждая Node в LinkedList?

Каждая Node в LinkedList хранит два поля:

Значение (сам элемент списка).

Ссылки на предыдущий и следующий узлы.

4. Можем ли мы добавить null в LinkedList?

Мы можем добавить null в LinkedList, так как null считается допустимым значением элемента списка.

5. Какая сложность вставки и удаления в LinkedList? какая у ArrayList?

Сложность вставки и удаления:

Вставка и удаление в середине списка (LinkedList): $O(1)$ (после поиска позиции).

Вставка и удаление в начале или конце списка (LinkedList): $O(1)$.

Вставка и удаление в середине списка (ArrayList): $O(n)$ (из-за сдвига элементов).

Вставка и удаление в начале или конце списка (ArrayList): $O(1)$.