

Список на основе массива

НАШИ ПРАВИЛА



Включенная камера



Вопросы по поднятой руке



Не перебиваем друг друга



Все вопросы, не связанные с тематикой курса (орг-вопросы и т. д.), должны быть направлены куратору



Подготовьте свое рабочее окружение для возможной демонстрации экрана (закройте лишние соцсети и прочие приложения)

ЦЕЛЬ

Понять принципы реализации методов списка на основе массива

ПЛАН ЗАНЯТИЯ

СПИСКИ

РЕАЛИЗАЦИЯ СПИСКА НА
ОСНОВЕ МАССИВА

ДОБАВЛЕНИЕ ЭЛЕМЕНТА В
СПИСОК

РАСШИРЕНИЕ СПИСКА

УДАЛЕНИЕ ЭЛЕМЕНТА

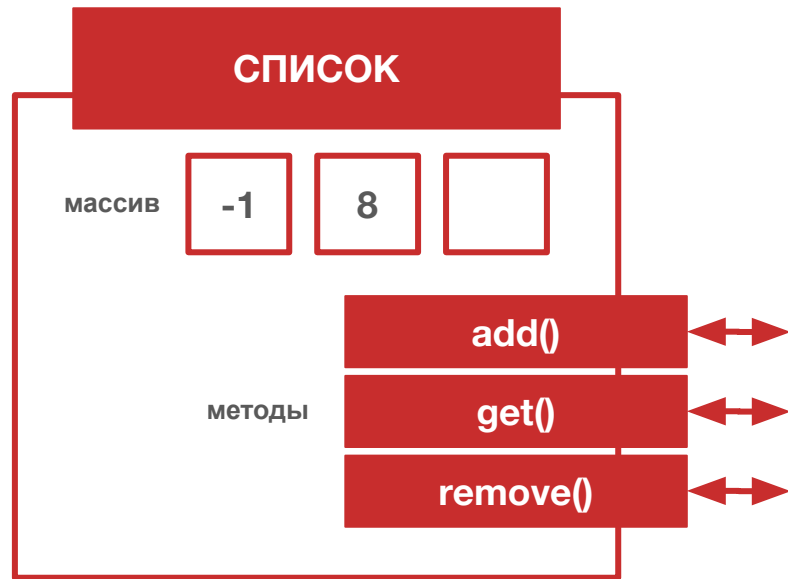
СПИСКИ

Список - абстрактный тип данных, который представляет собой коллекцию элементов с соответствующим набором операций:

- Добавить элемент
- Получить элемент
- Удалить элемент
- и т.д.

Списки и массивы имеют ряд сходств. Тем не менее массив - это структура данных, которая описывает способ хранения элементов, тогда как список - это не просто хранилище, а ассоциированный с этим хранилищем набор операций.

Списки обычно реализованы через классы, действующие как оболочки вокруг массивов. Эти классы предоставляют дополнительные функциональные возможности и контроль над организацией массива



СТАВИМ ПЛЮС, ЕСЛИ ПОНЯТНО



Отличие списков и массивов



Методы списка

РЕАЛИЗАЦИЯ СПИСКА НА ОСНОВЕ МАССИВА

Для реализации списка на основе массива необходимо определить начальный размер массива, который будет содержать будущие элементы и конструктор, инициализирующий этот массив.

Фактический размер массива и количество элементов списка будет отличаться. Необходимо определить поле **count**, которое будет показывать количество элементов в списке

0	1	2	3	4
7	2	-5		

Длина массива имеет значение **5**, при этом **count** должен быть равен **3**

```
class MyArrayList {  
  
    // начальный размер массива  
    private static int INITIAL_SIZE = 10;  
    // массив для хранения элементов  
    private int[] data;  
    // количество элементов списка  
    private int count = 0;  
  
    public MyArrayList() {  
        // создание массива  
        this.data = new int[INITIAL_SIZE];  
    }  
}
```

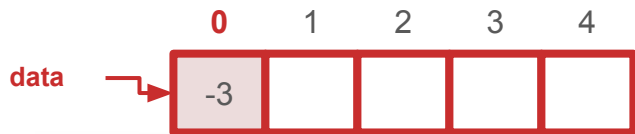
СТАВИМ ПЛЮС, ЕСЛИ ПОНЯТНО

■ Фактический размер массива и количество элементов списка

■ Поля списка на основе массива

ДОБАВЛЕНИЕ ЭЛЕМЕНТА В СПИСОК

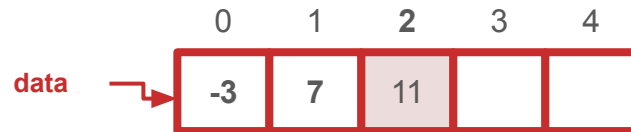
В простейшем случае мы можем добавить новый элемент в первую пустую позицию (в конец списка). Для этого заметим, что индекс первой пустой позиции всегда равен значению переменной `count`:



Значение `count` = 0. Новый добавляемый элемент `-3` попадает в `data[count]` или `data[0]`. Увеличиваем `count`



Значение `count` = 0. Новый добавляемый элемент `7` попадает в `data[count]` или `data[1]`. Увеличиваем `count`



Значение `count` = 2. Новый добавляемый элемент `11` попадает в `data[count]` или `data[2]`. Увеличиваем `count`

```
// реализация метода добавления
public void add(int element) {
    data[count] = element;
    count++;
}
```

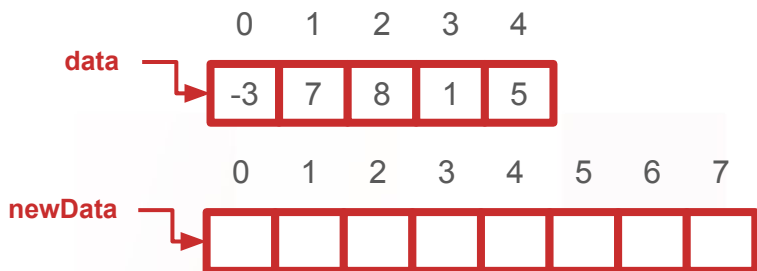
СТАВИМ ПЛЮС, ЕСЛИ ПОНЯТНО

■ Алгоритм добавления элемента в массив

■ Реализация добавления элемента в список

РАСШИРЕНИЕ СПИСКА

Если массив будет переполнен, необходимо предусмотреть расширение массива. Мы не можем изменить размер массива, для этого необходимо создать новый массив `newData` большего размера:



Далее все элементы старого массива следует скопировать в новый:



Осталось заменить ссылку `data` на новый массив, старый будет удален автоматически:



// обновленный метод добавления

```
public void add(int element) {
```

```
    // если массив заполнен
```

```
    if (count == data.length) {
```

```
        resize(); // “расширяем” его
```

```
    }
```

```
    // добавляем элемент
```

```
    data[count] = element;
```

```
    count++;
```

```
}
```

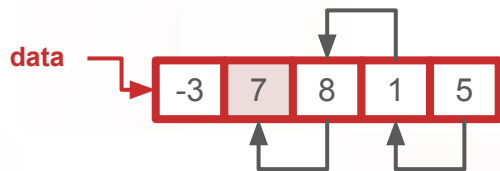
СТАВИМ ПЛЮС, ЕСЛИ ПОНЯТНО

■ Алгоритм “расширения” массива

■ Реализация добавления элемента с учетом расширения

УДАЛЕНИЕ ЭЛЕМЕНТА

При удалении элемента из списка по его индексу, необходимо сдвинуть все элементы влево на одну позицию и уменьшить count на 1:



СТАВИМ ПЛЮС, ЕСЛИ ПОНЯТНО

■ Алгоритм удаления элемента

■ Реализация удаления элемента

ПОИГРАЕМ ;)

Список

Список на основе массива


“Расширение” массива

ДОМАШНЕЕ ЗАДАНИЕ

Написать класс `ArrayList`, который будет содержать методы `add`, `removeByIndex`, `getByIndex`.

Варианты:

1. `CurrencyConverter` - используйте `ArrayList` для хранения `currencyRates` (не забудьте что `ArrayList` тут должен хранить тип `double`)
2. `Pokemon` - используйте `ArrayList` для хранения команд покемонов (не забудьте что `ArrayList` тут должен хранить тип `Pokemon`)
3. `L17` - используйте `ArrayList` для хранения пяти машин (не забудьте что `ArrayList` тут должен хранить тип `Car`)



Ваша новая IT-профессия – Ваш новый уровень жизни

Программирование с нуля в
немецкой школе AIT TR GmbH