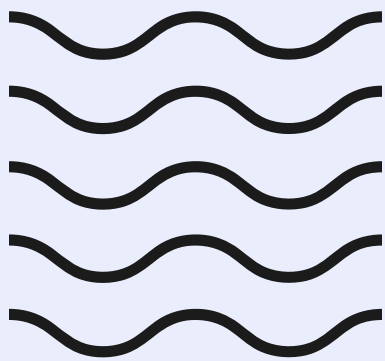


ArrayList



ПОДГОТОВИЛ: АЛИШЕР ХАМИДОВ



Основные моменты:



План лекции

1. Какие неудобства есть у массивов?
2. Как создать ArrayList?
3. Преимущества ArrayList
4. Как добавить элемент?
5. Как получить элемент?
6. Как заменить элемент?
7. Как удалить элемент?
8. Как удалить все элементы?
9. Циклы
10. Важные замечания





Какое самое большое неудобство массива?



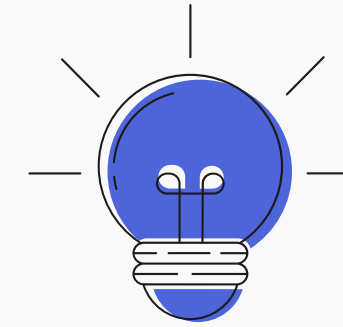
ArrayList - преимущества

resizable
(изменяемый
размер)

+

удобные методы

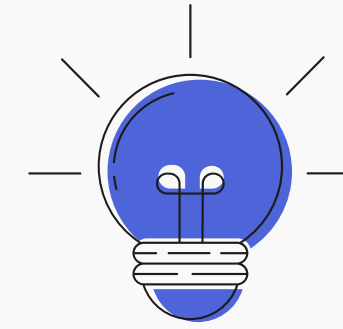
Как создать?



```
ArrayList<String> names = new ArrayList<>();
```

```
ArrayList<Тип хранимого значения> имя = new ArrayList<>();
```

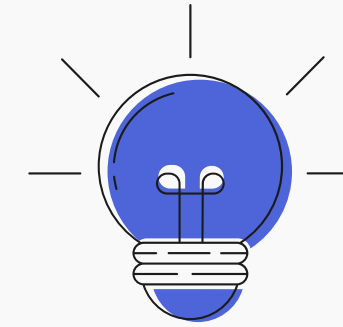
Как создать?



```
ArrayList<String> names = new ArrayList<>(10);
```

```
ArrayList<Тип хранимого значения> имя = new ArrayList<>  
    (вместимость исходного массива);
```

Как создать?



```
ArrayList<String> names = new ArrayList<>(previousNames);
```

(Создание из другого Array листа)

Как добавить элемент?

Чтобы добавить элемент используйте метод **add()**.

```
ArrayList<String> names = new ArrayList();  
names.add("Bob");  
names.add("John");  
System.out.println(names); // [Bob, John]
```


Как получить элемент?

Чтобы получить элемент из ArrayList нужно воспользоваться методом **get()** и передать в него индекс.

```
String name = names.get(0);  
System.out.println(name); // Bob
```

Как заменить элемент?

Чтобы заменить элемент в ArrayList нужно воспользоваться методом **set()** и передать в него индекс элемента, который мы хотим заменить и новое значение.

```
names.set(0, "Tim");  
System.out.println(names); // [Tim, John]
```

Как удалить элемент?

Чтобы удалить элемент в ArrayList нужно воспользоваться методом **remove()** и передать в него **индекс** элемента, который мы хотим удалить.

```
ArrayList<String> names = new ArrayList();  
names.add("Bob");  
names.add("John");  
names.remove(1);  
System.out.println(names); // [Bob]
```

Как ещё удалить элемент?

Чтобы удалить элемент в `ArrayList` нужно воспользоваться методом **`remove()`** и передать в него **элемент**, который мы хотим удалить.

```
ArrayList<String> names = new ArrayList();  
names.add("Bob");  
names.add("John");  
names.remove("Bob");  
System.out.println(names); // [John]
```

Как удалить все элементы?

Чтобы удалить все элементы из ArrayList нужно воспользоваться методом **clear()**.

```
ArrayList<String> names = new ArrayList();  
names.add("Bob");  
names.add("John");  
names.clear();  
System.out.println(names); // []
```

Как узнать сколько элементов?

Чтобы узнать количество элементов используется метод **size()**.

```
ArrayList<String> names = new ArrayList();  
names.add("Bob");  
names.add("John");  
System.out.println(names.size()); // 2
```

Как пройти по ArrayList циклом for?

```
ArrayList<String> names = new ArrayList();  
names.add("Bob");  
names.add("John");  
names.add("Ann");  
for(int i = 0; i < names.size(); i++){  
    System.out.println(names.get(i));  
}
```

Как пройти по ArrayList циклом for-each?

```
ArrayList<String> names = new ArrayList();  
names.add("Bob");  
names.add("John");  
names.add("Ann");  
for (String name : names) {  
    System.out.println(name);  
}
```

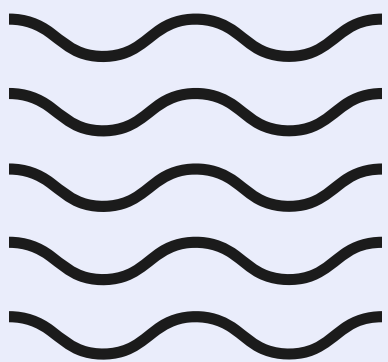

Важное замечание

Элементы ArrayList на самом деле являются **объектами**. В ✕ примерах выше мы создавали объекты класса String.

Вы же помните, что String в Java является объектом (не примитивный тип).

Чтобы использовать другие типы, например int, вам нужно воспользоваться подходящим классом-оберткой: Integer.

Для других типов используйте: Boolean для boolean, Character для char, Double для double, и т.д.





Документация

По ссылке ниже вы сможете ознакомиться с документацией и подробнее прочитать про все методы класса ArrayList

<https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html>

