JAVA
ARRAYLIST

Java **ArrayList** class uses a dynamic array for storing the elements. It is like an array, but there is no size limit. **We can add or remove elements anytime.** So, it is much more flexible than the traditional array. It is found in the **java.util package**. It is like the Vector in C++.

Important points about ArrayList

- Java ArrayList class can contain duplicate elements.
- Java ArrayList class maintains insertion order.
- Java ArrayList class is non-synchronized.
- Java ArrayList allows random access because the array works on an index basis.
- We can not create an array list of the primitive types, such as int, float, char, etc. It is required to use the required wrapper class in such cases. **Example -**

```
ArrayList<int> al = ArrayList<int>( ); // does not work
ArrayList<Integer> al = new ArrayList<Integer>( ); // works fine
```

Create ArrayList

```java
import java.util.ArrayList;

class Main {
  public static void main(String[] args){

    // create ArrayList
    ArrayList<String> languages = new ArrayList<>();

    // Add elements to ArrayList
    languages.add("Java");
    languages.add("Python");
    languages.add("Swift");
    System.out.println("ArrayList: " + languages);
  }
}
```

Basic Operations on ArrayList

The **ArrayList** class provides various methods to perform different operations on arraylists.

- **Add** elements
- **Access** elements
- **Change** elements
- **Remove** elements

1. Add Elements

To add a single element to the arraylist, we use the **add( ) method** of the ArrayList class.

```java
import java.util.ArrayList;

class Main {
  public static void main(String[] args){
    // create ArrayList
    ArrayList<String> languages = new ArrayList<>();

    // add() method without the index parameter
    languages.add("Java");
    languages.add("C");
    languages.add("Python");
    System.out.println("ArrayList: " + languages);
  }
}
```

2. Access Elements

To access an element from the arraylist, we use the **get( ) method** of the ArrayList class.

```java
import java.util.ArrayList;

class Main {
  public static void main(String[] args) {
    ArrayList<String> animals = new ArrayList<>();

    // add elements in the arraylist
    animals.add("Cat");
    animals.add("Dog");
    animals.add("Cow");
    System.out.println("ArrayList: " + animals);

    // get the element from the arraylist
    String str = animals.get(1);
    System.out.print("Element at index 1: " + str);
  }
}
```

3. Change Elements

To change elements of the arraylist, we use the **set( ) method** of the ArrayList class.

```java
import java.util.ArrayList;

class Main {
  public static void main(String[] args) {
    ArrayList<String> languages = new ArrayList<>();

    // add elements in the array list
    languages.add("Java");
    languages.add("Kotlin");
    languages.add("C++");
    System.out.println("ArrayList: " + languages);

    // change the element of the array list
    languages.set(2, "JavaScript");
    System.out.println("Modified ArrayList: " + languages);
  }
}
```

4. Remove Elements

To remove an element from the arraylist, we can use the **remove( ) method** of the ArrayList class.

```java
import java.util.ArrayList;
class Main {
  public static void main(String[] args) {
    ArrayList<String> animals = new ArrayList<>();

    // add elements in the array list
    animals.add("Dog");
    animals.add("Cat");
    animals.add("Horse");
    System.out.println("ArrayList: " + animals);

    // remove element from index 2
    String str = animals.remove(2);
    System.out.println("Updated ArrayList: " + animals);
    System.out.println("Removed Element: " + str);
  }
}
```

Jayesh Deshmukh

Follow

Like Comment Share Save