DAY-52

13 September 2023 11:05

```
package collectionpractice;

import java.util.ArrayList;

public class ALDemo
{

	public static void main(String[] args)
	{
		ArrayList a1 = new ArrayList();
		a1.add(1);
		a1.add(2);
		System.out.println("a1 arraylist : "+a1);

		ArrayList a2 = new ArrayList();
		a2.add(1);
		a2.add(2);
		System.out.println("a2 arraylist : "+a2);

		a1.addAll(1, a2);
		System.out.println(a1);
		a1.add(2, 3);
		System.out.println("after adding 3 at index 2"+a1);
		System.out.println("index of 1 in a1 is "+a1.indexOf(1));
		System.out.println(" last index of 1 in a1 is "+a1.lastIndexOf(1));
		System.out.println("object at index 3 is "+a1.get(3));
		a1.set(0, 10);
		System.out.println("after replacing object at index 0 with 10 "+a1);
		a1.remove(0);
		System.out.println("after removing object at index 0 "+a1);



	}

}
```

WAP TO PRINT ONLY INTEGER OBJECTS FROM THE ARRAY LIST.

```
package collectionpractice;

import java.util.ArrayList;

public class ALDemo
{
```

```
	public static void main(String[] args)
	{
		ArrayList a1 = new ArrayList();
		a1.add(3);
		a1.add("java");
		a1.add('a');
		a1.add(4);
		a1.add(true);

		for (int i = 0; i < a1.size(); i++)
		{
			Object obj = a1.get(i);
			if (obj instanceof Integer)
			{
				System.out.println(obj);

			}



		}


	}

}
```

QUESTIONS :

- WAP TO CREATE AN ARRAYLIST AND ADD OBJECTS TO IT. PRINT ONLY THE DOUBLE TYPE OF OBJECTS AND PRINT THE NUMBER OF CHARACTER OBJECTS.
- WAP TO ADD OBJECTS TO ARRAYLIST AND PRINT ONLY THE EVEN NUMBERS.

COLLECTIONS (C)

- IT IS A UTILITY CLASS WHICH CONTAINS STATIC METHODS USED TO PERFORM CERTAIN OPERATIONS RELATED TO THE COLLECTION OBEJCTS.
- THERE IS A METHOD CALLED AS Collections. shuffle(a); WHERE **a** IS LIST TYPE OF OBJECTSS
- GIVEN AN ARRAYLIST COLLECTION, SWAP THE OBJECTS AT A POSITION 2 & 3 Collections.swap(a, 2, 3);
- TO REVERSE THE ORDER OF THE OBJECTS IN THE ARRAY LIST. Collections.reverseorder(a); [SORT THE ARRAY IN THE DESCENDING ORDER]
- GIVEN THE ARRAYLIST CONVERTED INTO THE ARRAY

```
	Object[] a1 = a.toArray();

	For(int I = 0; I < a1.length; i++;)
	{
			System.out.println(a1[i]);
	}
```

- Create an ArrayList collection in such way that it accepts only integer objects [generic collections only homogeneous]

  ArrayList<Integer> a1 = new ArrayList();

- WAP TO SUFFLE THE OBJECTS IN ARRAY LIST

```
package collectionpractice;

import java.util.ArrayList;
import java.util.Collections;

public class ALDemo
{

	public static void main(String[] args)
	{
		ArrayList a1 = new ArrayList();
		a1.add(1);
		a1.add(0);
		a1.add(100);
		a1.add(2);
		a1.add(20);

		System.out.println("before shuffling"+a1);

		Collections.shuffle(a1);//used to shuffle objects in collections
		//randomly

		System.out.println("after shuffling "+a1);

	}

}
```

- WAP TO SWAP THE OBJECTS IN ARRAYLIST

```
package collectionpractice;

import java.util.ArrayList;
import java.util.Collections;

public class ALDemo
{

	public static void main(String[] args)
	{
		ArrayList a1 = new ArrayList();
		a1.add(1);
		a1.add(0);
		a1.add(100);
		a1.add(2);
```

```
            a1.add(20);

            System.out.println("before swapping "+a1);

            Collections.swap(a1, 2, 3);

            System.out.println("after swapping "+a1);

        }

    }
```

- WAP TO SORT THE OBJECTS IN ARRAY LIST

```
package collectionpractice;

import java.util.ArrayList;
import java.util.Collections;

public class ALDemo
{

    public static void main(String[] args)
    {
        ArrayList a1 = new ArrayList();
        a1.add(1);
        a1.add(0);
        a1.add(100);
        a1.add(2);
        a1.add(20);

        System.out.println("before sorting "+a1);


        Collections.sort(a1);
        System.out.println("after sorting "+a1);
    }

}
```

- WAP TO SORT THE OBJECTS IN ARRAYLIST IN DESCENDING ORDER

```
package collectionpractice;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;

public class ALDemo
{
```

```
        public static void main(String[] args)
        {
            ArrayList a1 = new ArrayList();
            a1.add(1);
            a1.add(0);
            a1.add(100);
            a1.add(2);
            a1.add(20);

            System.out.println("before sorting "+a1);


            Comparator<Object> b = Collections.reverseOrder();
            Collections.sort(a1, b);
            System.out.println("descending order "+a1);
        }

    }
```

- WAP TO CONVERT THE ARRAYLIST TO ARRAY

`

```
package collectionpractice;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;

public class ALDemo
{

    public static void main(String[] args)
    {
        ArrayList a1 = new ArrayList();
        a1.add(1);
        a1.add(0);
        a1.add(100);
        a1.add(2);
        a1.add(20);

        Object[] a2 = a1.toArray();
        for (int i = 0; i < a2.length; i++)
        {
            System.out.println(a2[i]);

        }
    }

}
```

ADVANTAGES OF ARRAYLIST

- IN ARRAYLIST OBJECTS ARE PLACES IN CONSECUTIVE MEMORY LOCATION AND IT ALSO

IMPLEMENTS RANDOMACESS(I) BECAUSE IF WHICH RETRIEVAL OPERATION IS FASTER.

- HENCE, IF OUR FREQUENT OPERATION US RETRIEVAL , THEN WE SHOULD GO FOR ARRAYLIST.

DISADVANTAGES OF ARRAYLIST

- SINCE ARRAYLIST OBEJCTS ARE STORED IN CONSECUTIVE MEMORY LOCATIONS , ANY ADDITION, DELETION IN BETWEEN ARRAYLIST WILL INTIALLY INVOLVED SEVERAL SHIFT OPERATIONS WHICH MAY BRING DOWN THE PERFORMANCE OF ARRAYLIST IF WE ARE USING FOR OUR APPLICATION.