

```

class ArrayUtils<T> {
    private T[] array;

    public ArrayUtils(T[] array) {
        this.array = array;
    }

    public void printArray() {
        for (T element : array) {
            System.out.print(element + " ");
        }
        System.out.println();
    }

    public int getLength() {
        return array.length;
    }

    public T getElement(int index) {
        if (index >= 0 && index < array.length) {
            return array[index];
        } else {
            throw new IndexOutOfBoundsException("Index out of range");
        }
    }

    public void setElement(int index, T element) {
        if (index >= 0 && index < array.length) {
            array[index] = element;
        } else {
            throw new IndexOutOfBoundsException("Index out of range");
        }
    }
}

```

```

public class ArrayAnyType {
    public static void main(String[] args) {
        Integer[] intArray = {1, 2, 3, 4, 5};
        ArrayUtils<Integer> intUtils = new ArrayUtils<>(intArray);

        intUtils.printArray(); // Output: 1 2 3 4 5

        String[] stringArray = {"Hello", "World"};
        ArrayUtils<String> stringUtils = new ArrayUtils<>(stringArray);

        stringUtils.printArray(); // Output: Hello World
    }
}

```

```

        Object[] objectArray = {"Hello", 0, 'x'};
    }
}

```

```

    ArrayUtils<Object> objUtils = new ArrayUtils<>(objectArray);

    objUtils.printArray(); // Output: Hello 0 x
}
}

```

We use this to create any type of array and can use arrayADT methods.

T indicate that type

Instead of defining what type of array in start we give user to freedom to implement ADT methods.

```
private T[] array;
```

```
Integer[] intArray = {1, 2, 3, 4, 5};
```

#we create integer array name intArray.

```
ArrayUtils<Integer> intUtils = new ArrayUtils<>(intArray);
```

#we create a intUtils object in ArrayUtils class and intArray is the argument.