

## UI Diagrams

In the case where a developer wants to make sure their code is correct with respect to array accesses they would do annotate as they code in the following ways:

### Using @IndexFor(a) with loops:

In order to show there is no possible out of bound exceptions this is how one would annotate a for loop over an array.

```
// this loop works properly no warnings
for(@IndexFor("arr") int i = 0; i < arr.length; i++){
    o = arr[i];
}

// due to an improper bounds check this will issue a warning for unsafe array access
for(@IndexFor("arr") int i = 0; i <= arr.length; i++){
    o = arr[i];
}

// this works fine
for(@IndexFor("arr") int i = arr.length - 1 ; i > -1 /* or >= 0*/; i--){
    o = arr[i];
}
```

### Using @IndexFor(a) with unknown values:

```
// if you are getting the index from somewhere else it works like this
int index = unknown /*result of another method, or user input, etc.*/;

o = arr[index]; // issues a warning: index could be out of bounds

// this works fine no warnings because bound were properly checked
if(index > -1 /* or >= 0*/ && index < arr.length){
    o = arr[index];
}
```

## Use Index Checker to find Non-Obvious out of bounds

Developer tries to do something with two arrays but doesn't check input properly. Because of this sometimes he runs into errors he doesn't know the cause of.

```
4 public static void main(String[] args){
5     int[] values = getvalues();
6     int valIndex = getinput();
7     if(valIndex < values.length){
8         System.out.println(values[valIndex]);
9     }
10 }
```

Problems @ Javadoc Console Terminal Git Staging Error Log wa

<terminated> UseCase2 [Java Application] /homes/iws/jsantino/Downloads/jdk1.8.0\_65/jre  
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: -1  
at proposal.UseCase2.main(UseCase2.java:8)

So he runs the checker to get warnings and find the error.

```
public static void main(String[] args){
    int[] values = getvalues();
    int valIndex = getinput();
    if(valIndex < values.length){
        System.out.println(values[valIndex]);
    }
}
```

Problems @ Javadoc Console Terminal Git Staging

Warning unsafe array access at line 8  
found values[@Length("values") i]  
expected values[@IndexFor("values") i]

The found type indicates that the value was not checked against zero, and therefore shows why the error happens and how to fix it.

## Using the Index Checker to find possible silent errors

A developer has code that uses a lot of arrays and they find that their output isn't what they expect. In order to make sure the errors are not with the array access they use the Index Checker to catch some types of silent errors.

He expected to see 59, the highest index in arr2 but for some reason he got 0.

```
8 public static void main(String[] args){
9     int[] arr = new int[30];
10    int[] arr2 = new int[60];
11    for(int i = 0; i < arr.length; i++){
12        arr[i] = i;
13    }
14    for(int i = 0; i < arr.length; i++){
15        arr2[i] = i;
16    }
17    System.out.println(arr2[arr2.length-1]);
18 }
19 }
20 }
```

Problems @ Javadoc Console Terminal Git Staging

<terminated> UseCaseUnAnnotated [Java Application] /homes/iws/

0

```
8 public static void main(String[] args){
9     int[] arr = new int[30];
0 int[] arr2 = new int[60];
1 for(@IndexFor("arr")int i = 0; i < arr.length; i++){
2     arr[i] = i;
3 }
4 for(@IndexFor("arr2")int i = 0; i < arr.length; i++){
5     arr2[i] = i;
6 }
7 System.out.println(arr2[arr2.length-1]);
8 }
9 }
10 }
```

Problems @ Javadoc Console Terminal Git Staging Error Log

```
1 Warning unsafe array access at line 15
2 found      arr2[@IndexFor("arr") i]
3 expected   arr2[@IndexFor("arr2") i]
```

The warning shows that the index is for the wrong array(he checked the wrong condition). He can now fix the mistake and get the correct behaviour.