

Array Idioms and Techniques

Idioms: typical usage of arrays

Overview

Idioms

1. processing all elements
2. selecting/counting elements
3. searching
4. finding max and min

Idiom 1: processing all elements.

- Printing all the elements of an array.
- To print every element in the array we must access each element. The most common way of doing this is with a for loop.
- The index of the for loop goes from 0 to array.length -1.

Ex 1. Printing an array

- To print the elements of array `count`, we use the following code

```
for(i = 0; i < count.length; i++)  
    System.out.println(count[i]);
```

- This works no matter what size the array is!

Ex 2. Reverse printing an array

- To print the elements of an array in reverse, we want the index to go in the opposite direction (from `length-1` to 0).

```
for(i = count.length - 1; i > 0; i--)  
    System.out.println(count[i]);
```

- Does this work?
- No, `i` doesn't reach the first element (`i` is never 0).

Every second element

- What happens if we want to print every second element in the array?
- This time, the index, `i`, must increase in steps of two.

```
for(i = 0; i < count.length; i = i + 2)  
    System.out.println(count[i]);
```

Every second element

- We could also use this technique

```
for(i = 0; i < count.length / 2; i++)
    System.out.println(count[i * 2]);
```

- as `i` goes from 0 to `count.length/2`, the index (`i * 2`) goes in steps of 2 from 0 to `count.length-2` (or `count.length-1` if `count.length` is odd).

Idiom 2. Selecting values

- Sometimes we only want to print certain values. For example, assume we have an array of student marks, and only want to print those who passed.

```
for(i = 0; i < mark.length; i++)
    if(mark[i] >= 40)
        System.out.println(mark[i]);
```

The for loop goes through the entire array and the if statement selects those which match a particular criteria (e.g. achieve a pass mark).

Counting values

- A related exercise is that of counting array elements. Here we want to see how many students achieved an honours grade (55%)

```
count = 0;
for(i = 0; i < mark.length; i++)
    if(mark[i] >= 55)
        count++;
```

- `count` is incremented once for each honour grade.

Idiom 3. Searching for an element

- Did any student get a mark of exactly 70? Stop searching if you find such a mark.

```
boolean found = false;
for(i = 0; i < mark.length; i++)
    if(mark[i] == 70)
    {
        found = true;
        break;
    }
if(found)
    print("student " + i + " got 70%");
else
    print("no student got 70%");
```

Checking values

- What happens if no student gets 70%?
- What happens if all students have 70%?

Checking values

- We want to see did any student got 90% or over. As before if one mark `>= 90%`, then we need look no longer; we can break out of the loop.

```
boolean found = false;
for(i = 0; i < mark.length; i++)
    if(mark[i] >= 90)
    {
        found = true;
        break;
    }
```

Idiom 4: finding max or min

- We want to find the student who got the highest mark. How do we do it?
- Will we have to check every element in the array?
- Where would we store the maximum value?
- How would you do it manually?
- Would you use a `for` loop? an `if` statement?

Finding max or min

- Have you seen a related problem?

```
int min(int a, int b)
{
    int min = a;
    if(b < min)
        min = b;
    return min;
}
```

```
int min(int a, int b, int c)
{
    int min = a;
    if(b < min)
        min = b;
    if(c < min)
        min = c;
    return min;
}
```

Finding the highest mark

- We need a loop to examine each element.
- We need an `if` statement to see if an element is greater than our best guess.

```
go through each element
if greater than guess
update guess
```

- What should we initialise `guess` to?

Finding the highest mark (code)

- Use a `for` loop to examine every element in the array.

```
int max = ?;
for(i = 0; i < mark.length; i++)
    if(mark[i] > max)
        max = mark[i];
```

- What should we initialise `max` to?

Finding the cheapest price

- Obviously if we have an array of computer prices and want to find the cheapest price, we'd use a similar technique

```
int min = price[0];
for(i = 1; i < price.length; i++)
    if(price[i] < min)
        min = price[i];
```

Summary

Idioms

1. processing all elements (for loop)
2. selecting/counting elements (for and if)
3. searching (for, if and break)
4. finding max and min (for, if and another variable)