Fixed-Size Collections

Arrays, for loop

suggested reading:

Textbook, Ch. 7

Fixed-size collections

- Sometimes the maximum collection size can be pre-determined.
- A special fixed-size collection type is available: an *array*.
- Unlike flexible collections, arrays can store object references or primitive-type values.
- Arrays use a special syntax.

The *weblog-analyzer* project

- Web server records details of each User's access.
- Supports analysis tasks:
  - Most popular pages.
  - Busiest periods.
  - How much data is being delivered.
- Analyze accesses by hour.

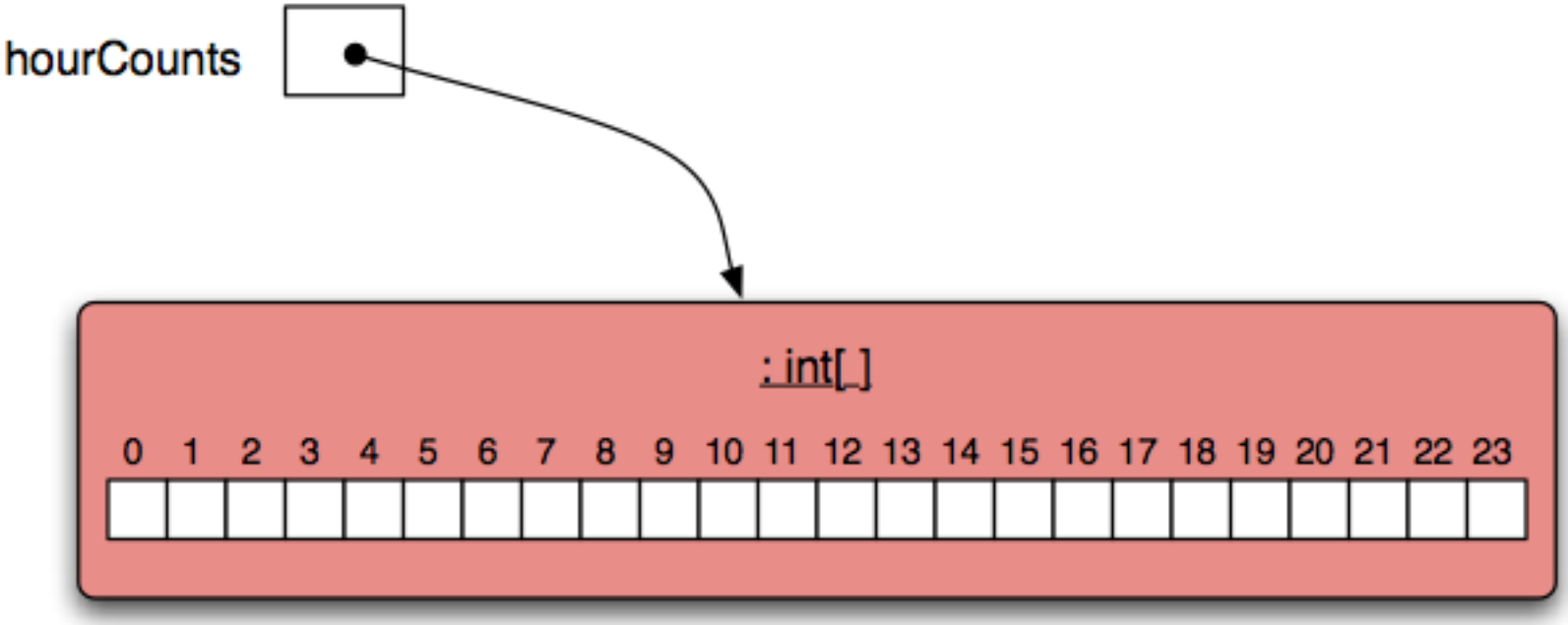
Creating an array object

```
public class LogAnalyzer
{
    private int[] hourCounts;
    private LogfileReader reader;

    public LogAnalyzer()
    {
        hourCounts = new int[24];
        reader = new LogfileReader();
    }
    ...
}
```

Array *variable declaration* — does *not* contain size

Array *object creation* — specifies size

The `hourCounts` array



Using an array

- Square-bracket notation is used to access an array element: `hourCounts[...]`
- Elements are used like ordinary variables.
- The target of an assignment:

```
hourCounts[hour] = ...;
```

- In an expression:

```
hourCounts[hour]++;
adjusted = hourCounts[hour] - 3;
```

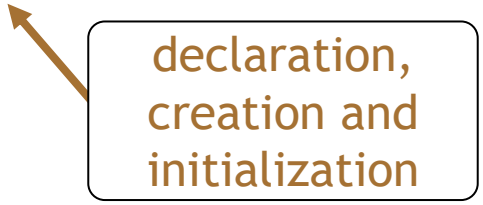
Standard array use

```
private int[] hourCounts;
private String[] names;

...

hourCounts = new int[24];

...

hourcounts[i] = 0;
hourcounts[i]++;
System.out.println(hourcounts[i]);
```

Array literals

- The size is inferred from the data.

```
private int[] numbers = { 3, 15, 4, 5 };
```



- Array literals in this form can only be used in declarations.
- Related uses require **`new`**:

```
numbers = new int[] {
    3, 15, 4, 5
};
```

Array length

```
private int[] numbers = { 3, 15, 4, 5 };

int n = numbers.length;
```

- NB: **`length`** is a field rather than a method!
- It cannot be changed – 'fixed size'.

The for loop

- There are two variations of the for loop, *for-each* and *for*.
- The for loop is often used to iterate a fixed number of times(*Definite iteration*).
- We need a variable inside the loop whose value changes by a fixed amount—typically increasing by 1—on each iteration.
- for-each loops can be used with arrays just like with other collections, but we lose our counter variable. We prefer using the old-style for loop because it is more concise.

For loop pseudo-code

General form of the for loop

```
for(initialization; condition; post-body action) {
    statements to be repeated
}
```

A Java example

for loop version

```
for(int hour = 0; hour < hourCounts.length; hour++) {
    System.out.println(hour + ": " + hourCounts[hour]);
}
```

Practice

- Given an array of numbers, print out all the numbers in the array, using a for loop.

```
int[] numbers = { 4, 1, 22, 9, 14, 3, 9};

for ...
```

for loop with bigger step

```
// Print multiples of 3 that are below 40.
for(int num = 3; num < 40; num = num + 3) {
    System.out.println(num);
}
```

Review

- Arrays are appropriate where a fixed-size collection is required.
- Arrays use a special syntax.
- For loops are used when an index variable is required.
- For loops offer an alternative to while loops when the number of repetitions is known.