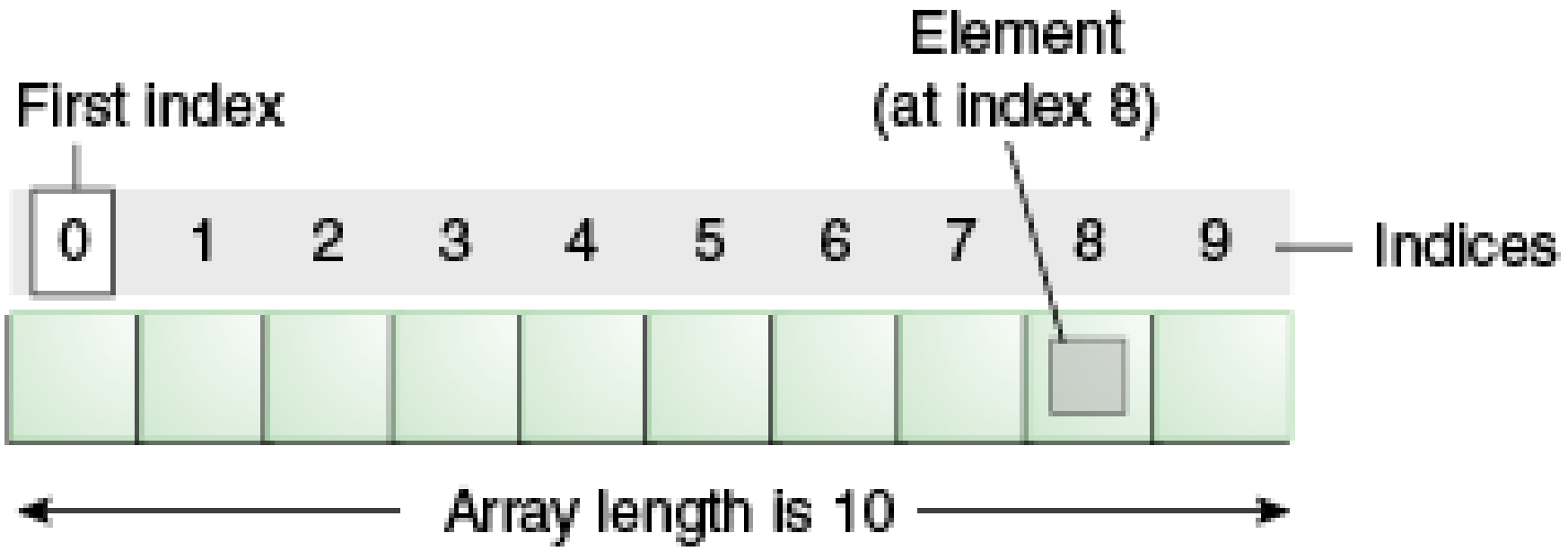
Chap. 7

Programming Arrays



Looping Through Arrays

Sect. 6.3

```
public static int sum(int[ ] array) {
        int tot = 0;
        if(array != null) {
                for (int i = 0; i < array.length; ++i) {
                        tot += array[ i ];
                }
        }
        return tot;
}
```

Make sure array is instantiated!

length is a public field in the array class, so we can read it

Run-Time Array Bounds Checking

- Use of array index is checked at run-time
- If array bound is exceeded, JVM "throws" an exception

```
int[] decade={0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
for(int i=0; i<=decade.length; i++) {
  System.out.println("Decade contains: " + decade[i]);
}
```

Produces:

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 10
        at ArrayBounds.main(ArrayBounds.java:7)
```

Enhanced “for” loop

Chap. 7.2

```
public static int sum(int … array) {
        int tot = 0;
        for (int t : array) {
                tot += t;
        }
        return tot;
}
```

like “foreach” in other languages

No index "i" available in the loop.

"t" is a copy of array value,
modifying "t" does not change the array.

Problem

- Given an array of bank accounts, return the bank account in the array with the minimum balance. If there is no account in the array with a minimum balance, return null.

- Strategy… loop through accounts, keeping track of the account with the lowest balance found so far.