

The background of the slide is a painting of a landscape. In the foreground, there is a field of yellow flowers, possibly rapeseed, with some blue irises in the lower left. In the middle ground, there are several green trees. In the background, a hillside features a church with a prominent tower and other buildings with red roofs. The sky is a mix of green and blue tones, suggesting a hazy or overcast day.

Arrays

J3

- Declaration and initialisation
- Indexing
- Arrays are mutable

Arrays

- An array is a data structure that stores a **sequence** of elements, all of the same type.
- In Java, the length of the array is fixed at creation.
- Elements of an array are accessed by their **index** (position in the sequence).
- In Java, arrays are indexed from zero (to length - 1).



- An array type name is made by adding a pair of square brackets (“[]”) to the end of the element type name:
 - For example, `int []`, `char []`, `String []`.
- Declare and initialise:

```
int[] numbers = {5, 6, 7, 8};
```
- Index: `numbers[0]` + `numbers[3]`
`numbers[i] > numbers[i - 1]`
 - `ArrayIndexOutOfBoundsException`
- Length: `numbers.length`



Example: Array functions

- Find the min/max element in an array.
- Compute the sum/average of a (numeric) array.
- Find a specific element
 - in an unordered array; or
 - in a sorted array (binary search).
- Find the median of a (numeric) array.



Arrays are mutable

- The “spaces” in an array are variables of the element type: they can be *assigned* as well as read.

```
numbers[numbers.length - 1] = 0;  
numbers[i] += 1;
```

- Example: Shuffle and sort.



N-dimensional arrays

- An array of arrays is declared by appending brackets ([]) to an array type.

```
int [] [] d = {  
    {1, 2, 3},  
    {0, 1, 2},  
    {0, 0, 1}  
};
```

- Index: `d[row][col]` – type `int`
`d[row]` – type `int []`

