Exercise 24: Comparing object arrays

Define a method
`boolean equals(Time[] tarray1, Time[] tarray2)`
which returns **true** if the two arrays `tarray1` and `tarray2` of references to `Time` objects are equal with respect to content.

Write a program that tests the method.

Exercise 25: Cinema

In this exercise, you should define a class `Cinema` that represents a cinema by using two-dimensional Boolean arrays – as done at the lecture.

The constructor should look like this:

- `Cinema(int[] seatsOnRow)`
  makes a new cinema object. The number of rows is equal to the number of elements in `seatsOnRow` and number of seats on each row is given by the numbers in the array `seatsOnRow`. Example: `Cinema(new int[]{6,5,4,3,3});` creates a `Cinema` object with six seats on row zero, five seats on row one, four seats on row two, and three seats on rows three and four.

The class should have the following methods:

`int rows()`
number of rows in the cinema.

`int seatsOnRow(int row)`
number of seats on row `row`.

`int seats()`
total number of seats in the cinema.

`int rowVacancies(int row)`
yields number of vacant seats on row `row`.

`int vacancies()`
the total number of vacant seats in the cinema.

`void book(int row, int seat)`
books seat `seat` on row `row`.

`void release(int row, int seat)`
cancels the booking of seat `seat` on row `row`.

`String toString()`
should return a character string showing which seats that are occupied (#) and which seats that are vacant (`O`). Example:

```
O O # # O O
O # # # #
O # # O
O O #
O O O
O
```

Also write a program `TestCinema` that tests `Cinema`.