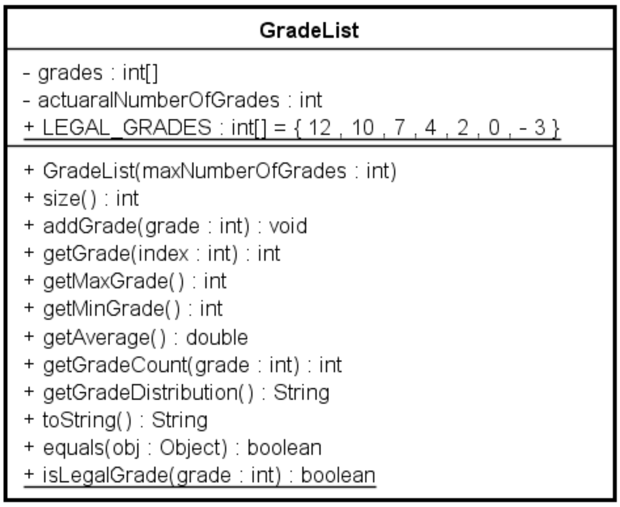
Exercise: GradeList, version 1



```
GradeList
- grades : int[]
- actuaralNumberOfGrades : int
+ LEGAL_GRADES : int[] = { 12 , 10 , 7 , 4 , 2 , 0 , - 3 }
+ GradeList(maxNumberOfGrades : int)
+ size() : int
+ addGrade(grade : int) : void
+ getGrade(index : int) : int
+ getMaxGrade() : int
+ getMinGrade() : int
+ getAverage() : double
+ getGradeCount(grade : int) : int
+ getGradeDistribution() : String
+ toString() : String
+ equals(obj : Object) : boolean
+ isLegalGrade(grade : int) : boolean
```

Create a new module and name it `GradeList_v1`. Add a new class called `GradeList` representing a list of grades in the Danish 7-grading scale.

The class `GradeList` has:

a) Two instance variables; `grades`, an `int` array for the grades and `acturalNumberOfGrades`, an `int` representing the number of grades added.
b) One public class constant (public `static final`) `LEGAL_GRADES` as an integer array with the legal grades in the Danish 7-grading scale.
c) A constructor taking the capacity (the maximum number of graded to add). The constructor creates the array of the given capacity and initialises the actual number of grades to zero.
d) A class method (`static` method) `isLegalGrade` returning `true` if the grade specified is a legal grade, i.e. can be found in the `LEGAL_GRADES` array.
e) A method `size` returning the actual number of grades added.
f) A method `addGrade` adding a grade in the array. Only add the grade if it is a legal grade and the array is not full.
g) A method `getGrade` returning the grade at the index specified.
h) A method `getMaxGrade` returning the maximum grade added to the array.
i) A method `getMinGrade` returning the minimum grade added to the array.
j) A method `getAverage` returning the average value of the grades added to the array.
k) A method `getGradeCount` returning the number of grades with the value as specified.
l) A method `getGradeDistribution` returning a string with a count for each of the legal grades.
m) A method `toString` returning a string with each of the grades shown in a comma separated list.
n) A method `equals` returning `true` if the argument is a `GradeList` object with the same size and same grades (identical array contents). *Note that you cannot directly use the template for method `equals` here.*

Implement a test class with a `main` method for your class `GradeList` (class `GradeList Test`). It is partly tested using these statements:

```
GradeList list = new GradeList(10);
list.addGrade(12);
```

```
        list.addGrade(-3);
        list.addGrade(7);
        list.addGrade(7);
        list.addGrade(12);
        list.addGrade(10);
        list.addGrade(0);
        list.addGrade(0);
        list.addGrade(4);
        list.addGrade(7);

        System.out.println("Grades: " + list);
        System.out.println("Average: " + list.getAverage());
        System.out.println("Grade 12 count: " + list.getGradeCount(12));
        System.out.println("Distribution: " + list.getGradeDistribution());
```

```
// OUTPUT
Grades: {12, -3, 7, 7, 12, 10, 0, 0, 4, 7}
Average: 5.6
Grade 12 count: 2
Distribution: {12 (2), 10 (1), 7 (3), 4 (1), 2 (0), 0 (2), -3 (1)}
```