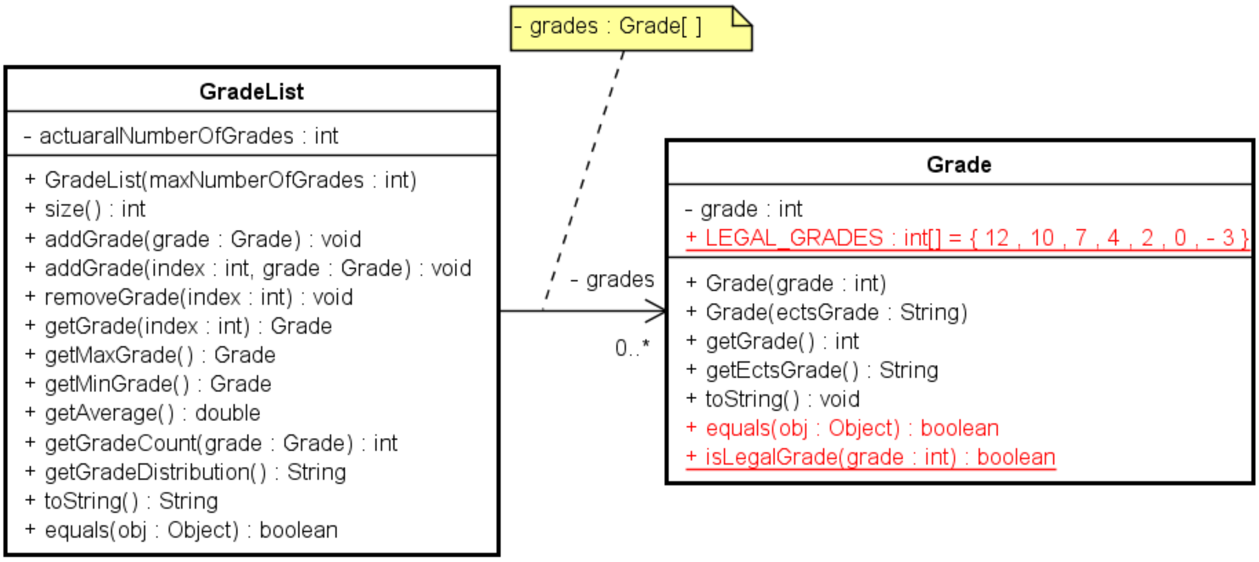
Exercise: GradeList, version 3


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

Create a new module `GradeList_v3` and copy the files from `GradeList_v2`. If you have an exercise with a `Grade` class then also copy this into the module.

Notes to class `Grade`:

a) Instance variable, constructors, getter for grade, a method returning the ECTS grade (using a switch) and a toString method as in the class diagram above

b) Add a public class constant (`public static final`) `LEGAL_GRADES` as an integer array with the legal grades in the Danish 7-grading scale. Simply move it from class `GradeList`.

c) Add a `static` method `isLegalGrade` returning `true` if the grade specified is a legal grade, i.e. can be found in the `LEGAL_GRADES` array. Simply move it from class `GradeList`.

d) Update constructor (and/or set method) to call the `static` method instead of making the checks.

e) Add a method `equals` returning `true` if the argument is a `Grade` object with the same grade value. *Note: use the template for method equals.*

f) It is ok (but not a must) to add another `static final` variable for the legal ECTS grades and a boolean method checking for legal ECTS grades.

Modify class `GradeList` with the following changes:

g) Change the type of instance variable `grades` to `Grade[]`.

h) Delete the constant `LEGAL_GRADES`

i) Delete method `isLegalGrade`

j) Change the constructor to create a `Grade` array instead of an `int` array.

k) Modify both versions of methods `addGrade` to take an argument of type `Grade` instead of `int`. Don't check for legal grades (because a `Grade` object is already checked)

l) Modify method `getGrade` to return the grade of type `Grade`.

m) Modify method `getMaxGrade` returning the maximum grade added to the array.

n) Modify methods `getMaxGrade`, `getMinGrade`, `getAverage` such that you get the integer value of the grade in your calculations (i.e. `grades[i].getGrade()`). Further, methods `getMaxGrade` and `getMinGrade` now returns a `Grade` object, i.e. you have to create this object at the end just before you return.

o) Modify method `getGradeCount` to use `equals` (in `Grade`) when comparing `Grade` object (using == will not work)

p) Modify method `getGradeDistribution` accordingly.
Note that you can access the `LEGAL_GRADES` array in class `Grade` this way:

```
Grade.LEGAL_GRADES[i]
```

Also note that you can call method `getGradeCount` this way:

```
getGradeCount(new Grade(Grade.LEGAL_GRADES[i]))
```

q) Modify method `equals` (in `GradeList`) to use `equals` (in `Grade`) when comparing `Grade` object (using == or != will not work). You could make an if statement this way:

```
if (!grades[i].equals(other.grades[i]))
{
  return false;
}
```

Update your test program `GradeListTest` such that you test all methods.