Exercises:

1. Circle
   - Data fields:
     - radius: int
     - color: String
   - Methods:
     - getArea(): double, returns the area of the circle
     - toString(): String, returns a string representing the circle object with its area and color.

2. Employee
   - Data Fields:
     - <u>employees: Employee[8]</u>
     - <u>count: int</u>
     - id: int = -1
     - name: String = “Default Employee”
     - salary: double = 23217.53
     - level: int = 0
   - Methods:
     - Constructor that takes *name, salary,* and *level*. Capitalizes the first character of the name and rest to lower case. Generates a 4-digit unique id based on the level. If the level is 2, generate a **random** 4-digit id that starts with 2, for example 2143.
     - Accessors for *id, name, salary* and *level*
     - Mutators for *name* and *salary*
     - promote(e: Employee): None, a static method that takes an *Employee* instance and increases its level by one.
     - demote(e: Employee): None, a static method that takes an *Employee* instance and decreases its level by one.
     - generateId(level: int): int, a private static method that generates a random 4-digit unique id according to level and returns it.
     - doWork(): None, a dummy method that demonstrates behavior of instance. Display “[this] done work”
     - toString(): String, a method that represents the instance. Should display name, id, salary, and level each in a new line and tabbed in.

3. Write a class named *Student* that has three fields: *name*, *grade*, and *courses*, and a constructor that takes the *name* and *grade* as parameters and assigns them to the fields. The courses field should be initialized as an empty array list of strings. The class should also have methods named addCourse() and removeCourse() that take a course name as a parameter and add or remove it from the courses list, respectively. The class should also have a method named printCourses() that prints the courses in the list, separated by commas.